

PRAIRIE VIEW A&M UNIVERSITY
ROY G. PERRY COLLEGE OF ENGINEERING
DEPARTMENT OF Electrical and Computer ENGINEERING
ELEG 4330-P01 Introduction to Digital Design

Fall 2021

I. COURSE DESCRIPTION IN ABET FORMAT

II ASSESSMENT RESULTS

Program Outcomes Assessed in Course

Assessment Summary of Class Semester Averages for Outcomes

End of Semester Course Outcomes Assessment Reports

II. Midterm

Summary of Topics covered by Midterm

(Summary of topics can be a separate page or include on the assignment itself)

Above Average (Graded original work)

Average (Graded original work)

Below Average (Graded original work)

III. PROJECT 1

Project 1 description

Original Quiz Questions & Master Solution/grading scheme used

Above Average (Graded original work)

Average (Graded original work)

Below Average (Graded original work)

IV. PROJECT 2

Project 2 description

Above Average (Graded original work)

Average (Graded original work)

Below Average (Graded original work)

V. FINAL EXAM

Summary of Topics covered by FINAL EXAM

Above Average (Graded original work)

Average (Graded original work)

Below Average (Graded original work)

Course Description in ABET Format

1.	Course number and name:	ELEG 4330 Introduction to Digital Design
2.	Credits and contact hours:	3 credit hours; 5 contact hours * Subject Area: Engineering Topics
3.	Instructor's or course coordinator's name:	Dr. Xiangfang (Lindsey) Li
4.	Text book: Title:	Digital Design: With an Introduction to the Verilog HDL, VHDL, and System Verilog
	Author:	Morris Mano & Michael Ciletti,
	Publisher, Year & Edition:	6th Edition; 2017 Publisher: Pearson
	a. Other Supplemental material	
5.	Specific course information	
	a. Brief description of the content of the course (catalog description)	(3-0) Credit 3 semester hours. The use of hardware description language and automated synthesis in design. hierarchical and modular design of digital systems. Control logic, synchronous and asynchronous sequential circuit design. Programmable logic devices and field programmable gate arrays. Circuit simulation for design verification and analysis. Timing-oriented design.
	b. Prerequisites or Co-requisites	<u>ELEG 2311: Logic Circuits</u> <u>ELEG 33073: Microprocessor System Design</u>
	c. ** Indicate whether a required, elective, or selected elective (as per definition provided) course in the program	Required for Computer Engineering
6.	a. Specific goals for the course specific outcomes of instruction (e.g. The student will be able to explain the significance of current research about a particular topic.) <i>Students will be able to :</i>	b. Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course. <i>The specific goals for the course specific outcomes listed on the left column will address the following student learning outcomes:</i>
	1 The course is a continuation of ELEG 3063 (Logic Circuits). Throughout the course basic concepts are introduced by way of examples that involve digital	Outcome 2: An ability to apply engineering design to produce

	systems design, hardware structures and computer simulation systems.	solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factor
	2. Provide experience in system design and simulation using programming gate arrays.	
	3. Students will be exposed to the design, analysis, programming and demonstration of functional digital systems.	
7	Brief list of topics to be covered	Review of Boolean algebra and logic gates (3 hours) Gate level Minimization (3 hours) Combinational logic (6 hours) Synchronous Sequential Logic Latches and Flip-Flops (4.5 hours) Analysis of Clocked Sequential Circuits (4.5 hours) Synthesis of Sequential Circuits (4.5 hours) State reduction and assignment (1.5 hours) Design procedure of Sequential Circuits (4.5 hours) Registers (4.5 hours) Counters (4.5 Hours) Memory and Programmable Logic (6 hours)

* For all Engineering programs, indicate subject area from the following: **Math, Basic science, Engineering topic or Other**

** **Required** courses are required of all students in the program, **elective** courses (often referred to as open or free electives) are optional for students, and **selected elective** courses are those for which students must take one or more courses from a specified group

Additional Course Syllabus Information

COURSE: ELEG 4330 P01 CRN: 17884
CLASS SCHEDULE: T & R (12:30 pm – 1:50 pm), Face to Face
LOCATION: New Electrical Engr Bldg 115
OFFICE EE building 336
Office Hours: W: 1pm-3pm T & R: 9:00am - 12:00pm; **Or by appointment**

GRADING POLICY:

The following grading policy will be used as a guide to determine academic performance:

Attendance/quiz(s)	15%
Homework	10%
Midterm	25%
Project(s)/Assignment(s)	25%
<u>Final Exam</u>	<u>25%</u>
Total	100%

Grading Criteria and Conversion:

A = 85 or more

B = 75 - 84

C = 65 - 74

D = 55 - 64

F = 54 or below

Any deviation from the above grading policy will be discussed in advance with the class. Such deviations will be considered only in situations to provide learning enrichment opportunities for the entire class.

Course Procedures or Additional Instructor Policies

Taskstream

Taskstream is a tool that Prairie View A&M University uses for assessment purposes. One of your assignments may be required to be submitted as an "artifact," an item of coursework that serves as evidence that course objectives are met. If applicable, more information will be provided during the semester by your department, but for general information, you can visit Taskstream via the link in eCourses.

ON-LINE RESOURCES:

eCourses (<http://ecourses.pvamu.edu/>) will be used to disseminate course materials and organize discussions about the course to help improve the learning experience of the students.

HOMEWORK POLICY:

Homework and design problems will be assigned to complement each unit of instruction. Students will be required to turn in or demonstrate each assignment. Homework has to be handed on time. Late homework will **not** be accepted unless due to acceptable reasons (as defined by

University policy).

PROJECT/ASSIGNMENT POLICY:

There will be assigned project(s)/assignment. The project(s) will help the students to absorb the lectures through an interactive approach. Students are required to hand in the project reports.

QUIZ POLICY:

There will be multiple quizzes throughout this course to motivate students to follow the instruction closely. The quizzes will be administered in the beginning of the class. There will be no make up quiz if you are absent or late for the class. Make up quiz only apply to excused absence, student must contact instructor immediately for scheduling make up time **within one week of the missing quiz**.

EXAM POLICY:

There is one comprehensive exam, and there will be quizzes throughout the course. Every student must take all exam/quizzes on the assigned dates. Any student who misses an exam without a valid excuse will automatically receive zero for that exam. Make-up exams will be administered in accordance with university policy.

ETHICS:

Cheating or plagiarism on assignments or exams will not be tolerated. Proven cases of ethical violations will result in a zero for the assignment/exam and possibility of further disciplinary actions in accordance with university policies.

CLASS ATTENDANCE:

See the University Class Attendance Policy, Effective September 1, 1998. PVAMU requires regular class attendance. Attending all classes supports full academic development of each student. Excessive absenteeism may result in a student's course grade being reduced to a grade of "F". Accumulation of one week of unexcused absences constitutes excessive absenteeism.

STUDENT ACADEMIC APPEAL PROCESS

Authority and responsibility for assigning grades to students rests with the faculty. However, in those instances where students believe that miscommunication, errors, or unfairness of any kind may have adversely affected the instructor's assessment of their academic performance, the student has a right to appeal by the procedure listed in the Undergraduate Catalog and by doing so within thirty days of receiving the grade or experiencing any other problematic academic event that prompted the complaint.

COLLEGE OF ENGINEERING TEXTBOOK POLICY

Students must acquire the textbook that is listed as "required" on the course syllabus. The textbook must be acquired by the 10th class day. Students are not allowed to share textbooks with other students who are currently registered in the same class. Failure to acquire (or show proof of purchase) the required textbook by the 10th class day will result in the student being administratively dropped from the course. The University will assess financial obligations for the course to the student as with any other dropped class according to the fee schedule. In addition, your financial aid may be affected by the subsequent registration action(s). Go to <http://www.pvamu.edu/pages/195.asp> for the Roy G. Perry College of Engineering Textbook Policy.

CONDUCT:

1. Students will conduct themselves in a manner that is respectful to their fellow classmates
2. No cell phones or other electronic devices during exams and tests.
3. Students are NOT allowed to wear caps/hats in class

AMERICANS WITH DISABILITY (ADA) STATEMENT:

Students with disabilities who believe that they may need an academic adjustment in this class, are encouraged to contact the Office of Disability Services at (936) 857-2610/2620 as soon as possible. Once you have received a letter of adjustment from the office, kindly make an appointment with me to discuss appropriate adjustments for this class.

University Rules and Procedures**Disability statement (See Student Handbook):**

Students with disabilities, including learning disabilities, who wish to request accommodations in class should register with the Services for Students with Disabilities (SSD) early in the semester so that appropriate arrangements may be made. In accordance with federal laws, a student requesting special accommodations must provide documentation of their disability to the SSD coordinator.

Academic misconduct (See Student Handbook):

You are expected to practice academic honesty in every aspect of this course and all other courses. Make sure you are familiar with your Student Handbook, especially the section on academic misconduct. Students who engage in academic misconduct are subject to university disciplinary procedures.

Forms of academic dishonesty:

1. Cheating: deception in which a student misrepresents that he/she has mastered information on an academic exercise that he/she has not mastered; giving or receiving aid unauthorized by the instructor on assignments or examinations.
2. Academic misconduct: tampering with grades or taking part in obtaining or distributing any part of a scheduled test.
3. Fabrication: use of invented information or falsified research.
4. Plagiarism: unacknowledged quotation and/or paraphrase of someone else's words, ideas, or data as one's own in work submitted for credit. Failure to identify information or essays from the Internet and submitting them as one's own work also constitutes plagiarism.

Nonacademic misconduct (See Student Handbook)

The university respects the rights of instructors to teach and students to learn. Maintenance of these rights requires campus conditions that do not impede their exercise. Campus behavior that interferes with either (1) the instructor's ability to conduct the class, (2) the inability of other students to profit from the instructional program, or (3) campus behavior that interferes with the rights of others will not be tolerated. An individual engaging in such disruptive behavior may be subject to disciplinary action. Such incidents will be adjudicated by the Dean of Students under nonacademic procedures.

Sexual misconduct (See Student Handbook):

Sexual harassment of students and employers at Prairie View A&M University is unacceptable and will not be tolerated. Any member of the university community violating this policy will be subject to disciplinary action.

Attendance Policy:

Prairie View A&M University requires regular class attendance. Excessive absences will result in lowered grades. Excessive absenteeism, whether excused or unexcused, may result in a student's course grade being reduced or in assignment of a grade of "F". Absences are accumulated beginning with the first day of class.

Student Academic Appeals Process

Authority and responsibility for assigning grades to students rests with the faculty. However, in those instances where students believe that miscommunication, errors, or unfairness of any kind may have adversely affected the instructor's assessment of their academic performance, the student has a right to appeal by the procedure listed in the Undergraduate Catalog and by doing so within thirty days of receiving the grade or experiencing any other problematic academic event that prompted the complaint.

COVID-19 Campus Safety Measures

To promote public safety and protect students, faculty, and staff during the coronavirus pandemic, PVAMU has adopted policies and practices to limit virus transmission.

- **Self-monitoring** - Students should follow CDC recommendations for self-monitoring. Students who have a fever or exhibit symptoms of COVID-19 should participate in class remotely and should not participate in face-to-face instruction.
- **Face Coverings** - Face coverings (cloth face covering, surgical mask, etc.) are recommended in classrooms, teaching laboratories, common spaces such as lobbies and hallways, public study spaces, libraries, academic resource, and support offices, and outdoor spaces where 6 feet of physical distancing is challenging to maintain reliably.
- **Physical Distancing** - Physical distancing should be maintained between students, instructors, and others in course and course-related activities where possible.
- **Personal Illness and Quarantine** - Students required to quarantine are to participate in courses and course-related activities remotely and must not attend face-to-face course activities. Students should notify their instructors of the quarantine requirement. Students under quarantine are expected to participate in courses and complete graded work unless they have symptoms that are too severe to participate in course activities. Students experiencing personal injury or illness that is too severe for the student to attend class qualify for an excused absence. To receive an excused absence, students must provide appropriate documentation to the Office for Student Conduct, studentconduct@pvamu.edu.

End of Semester Course Assessment Report

This summary contains the following Information

1. Performance Statistics

- Total number of students in class
- The class average performance (**stated as %**) in each course outcome assessed in this class and the **two previous times course was assessed**.
- The acceptable class average for the class in each outcome area as set by your Department. **You cannot use any number of your choice**
- Percentage of students who scored below the expected average in each outcome.

2. Implementation Summary

- Brief summary of plans implemented during the semester based on last assessment report

3. Perceived Problems

- Instructor's **critical evaluation** of perceived problems that affected students' performance. **Realistic problems should be identified throughout the semester and summarized in this section of the report.**

4. New Plans for Addressing Problems

- Instructor's plans for addressing the perceived problems the next time the class is taught. **These should be specific enough for another instructor to implement. Most of these should be things that, you, the instructor can implement to help students to improve. Be creative.**

5. Overall Trend over last three assessment periods

- Instructors summary of overall performance trend for each outcome measured over the last three periods.

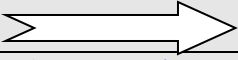
5. Were Expectations Met?

- Answer **YES** if the **class average was above the expected average** and the **stated percentage of students** scored at or above the expected average, else answer NO.

Table 2.1 END OF SEMESTER COURSE OUTCOME ASSESSMENT REPORT
ELEG 4330-P01 Introduction to Digital Design
Fall 2021 SEMESTER

Report Prepared by: Dr. Xiangfang Li

Report Date: Dec 19, 2021

Semester	Analysis Type	Number of Students	Outcome 2: An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factor			
			Expected Average	% Students Above Expected Average	Class Average	% Students Above Expected Average
Fall 2021	Direct	17	75	70%	81.6	82%
Fall 2020	Direct					
Implementation Summary		Two projects were given and the project requirement were prepared and made available through e-Courses. The projects' requirements and report writing were explained in the class.				
Perceived Problems		The projects given are tightly related to the materials studied in the class. The objectives of the project are to give the students opportunity to digest, design, and test their projects. Because of COVID-19, some students' attendance could not be guaranteed. This "introduction to digital design" is a very important course to introduce "digital world" journey to the students. The initial background and starting point of the students in this class are a little bit weak. Hence, a systematic methodology is adopted to explain the materials in order to help students to understand the background and big picture, and systematically connect different chapters/modules together.				
Plans for Addressing Problems		Face to face instruction was adopted for the course, at the same time, zoom session is open for the students who could not attend. Majority of the students attend the course regularly in person, hence, students' feedbacks are inquired regularly and the progress of the course are dynamically adjusted in order to better deliver the material. Multiple ways are adopted to encourage the students to actively involve in the course, regularly homework and quizzes are given to push the students to keep up the pace. The instructor is very satisfied to see the students' progress. Two projects are given to the students, and majority of the students finished the projects successfully.				
Overall Trend over Periods		Overall trend is improving, the students learning results are improving along the semester.				
Were Expectations Met?	Average ≥ 70%? and ≥ 70% students avg. ≥ 70%?		YES			

ELEG 4330-P01 Introduction to Digital Design

Fall 2021

Title of Assignment: Midterm

TOPICS COVERED IN Midterm

Test 1 covered the following topics:

1. Binary arithmetic and Boolean algebra
2. Boolean function simplification, don't care situation, Sum-of-product form.
3. Combinational logic, using available Combinational logic block to design Combinational circuit.
4. Prime implicants and Essential prime implicants.
5. Design combinational logic using universal gates.
6. Design a combinational logic given a practical problem

ELEG 4330-P01 Introduction to Digital Design
Fall 2021

Midterm

ELEG 4330 Introduction to Digital Design
Department of Electrical & Computer Engineering
PVAMU

Midterm Exam

Fall 2021

Name: _____

1	15	
2	10	
3	10	
4	10	
5	10	
6	10+3	
7	12	
8	15	
9	10	
Total		

Instructions:

Please keep your middle steps in the exam paper. Only the final result will NOT give you full credit.

You have 80 minutes to complete the exam. This exam counts for 25% of your course grade.

(15 pts) 1. Binary arithmetic and Boolean algebra

(3 pts) (1.1). Convert unsigned binary number to decimal number

$(110110.01)_2$

(3 pts) (1.2). Convert the Hexdecimal number to binary number

B3F0

(3pts) (1.3) . Please find the **complement** of the following function and simplify it.

$$F = A'B(C+D') + E$$

$$F' =$$

(6 pts) (1.4). Please use the Boolean Algebra Theorems to simplify the following boolean expression to a minimum number of literals.

$$F_1 = (BC' + A'D)(AB' + CD')$$

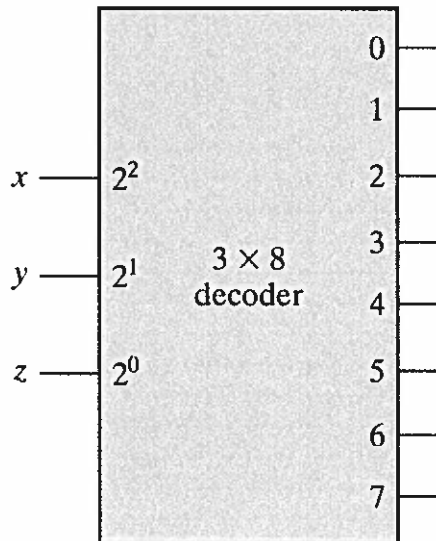
$$F_2 = (x(y'z' + yz))'$$

(10 pts) 2. Decoder

Use a decoder and external gate design the combination circuits (F_1 and F_2) defined by

$$F_1(x,y,z)=xy+yz$$

$$F_2(x,y,z)=xyz+y'z$$



(10 pts) 3. Simplify the following Boolean Function F , together with the **Don't-Care** condition d , and then please express the function in simplified Sum-of-Product form:

$$F(A,B,C,D)=\sum_m(2,3,4,7,10,12)$$

$$d(A,B,C,D)=\sum_m(0,6,8,11,13)$$

(10pts) 4. Prime implicants and Essential prime implicants

(a) Specify the following 4-variable boolean function in **simplified SOP form**.

(b) Please list the **Essential prime implicants**.

$$F(A,B,C,D) = \sum_m (0, 2, 5, 7, 8, 10, 12, 13, 14, 15)$$

(10 pts) 5. Design a three-input, one-output minimal two-level gate combinatorial circuit using **only NAND** (you may use NOT gates if needed) gates that has a "1" output when only the majority of its inputs are "1", otherwise, output is "0".

(3pt) a. Draw the truth table

(3pt) b. Minimize the function using K-Map

(4pt) c. Implement the logic using **only NAND** gates

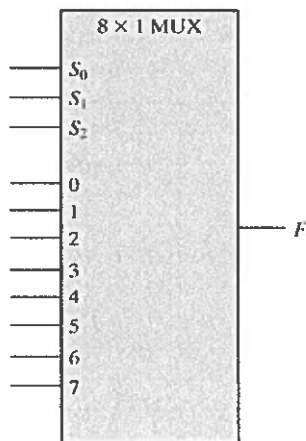
(10+3 pts) 6. Please design a logic to compare the magnitude of two 3-bit unsigned binary number (A, B): denoted by $A_3A_2A_1$ and $B_3B_2B_1$. Please show the boolean equations of how to determine whether one number is greater than ($A > B$), less than ($A < B$), or equal ($A = B$) to the other number.

Please **show the design process**, and demonstrate the **boolean expressions** and **logic circuit** (3 bonus points for correct logic circuit).

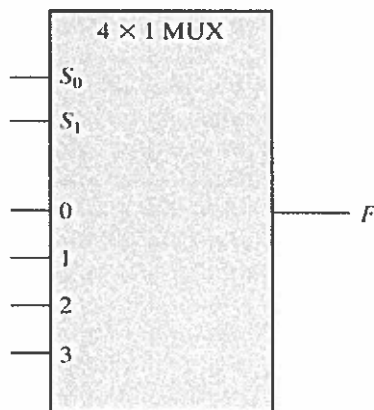
(12 pts) 7. Given the truth table below, please implement the truth table using a multiplexer.

(6 pts) (7.1) please using an 8-to-1 multiplexer to implement the truth table.

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



(6 pts) 7.2 please using a 4-to-1 multiplexer to implement the truth table. Briefly explain the design process.

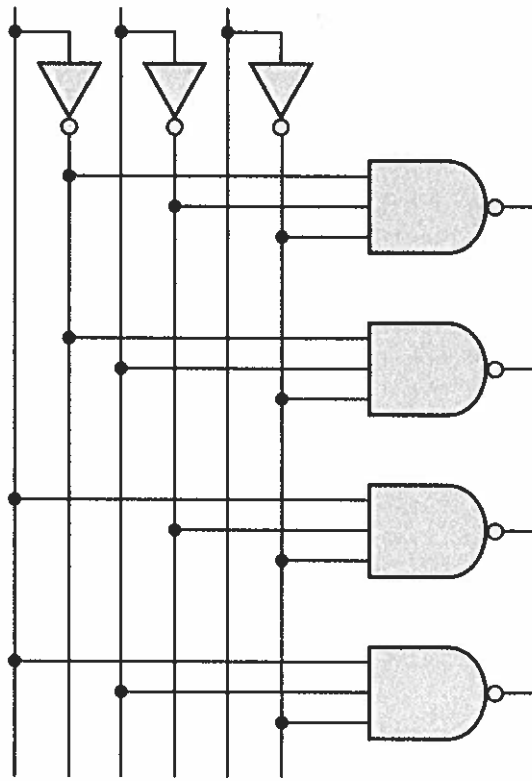


(15 pts) 8. Design a combinational circuit with three inputs x , y , and z and three outputs A , B , and C . When the binary input is 0, 1, 2, or 3, the binary output is one greater than the input. When the binary input is 4, 5, 6, or 7, the binary output is two less than the input.

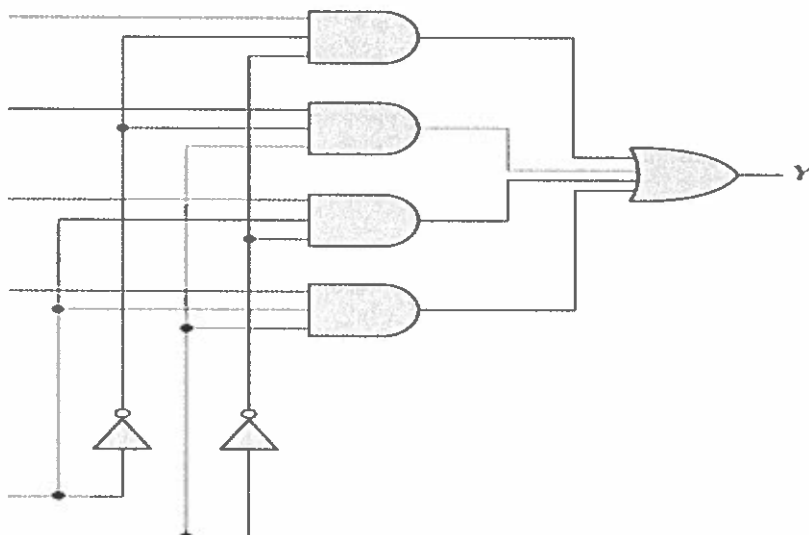
Please demonstrate the design process (truth table, K-map, and finally the boolean expression and logic circuits).

(10 pts) 9. Please analyze the logic, describe the function, and write the truth table/function table.

(5pt) 9.1



(5pt). 9.2



ABOVE AVERAGE

ELEG 4330 Introduction to Digital Design
Department of Electrical & Computer Engineering
PVAMU

Midterm Exam

Fall 2021

Name: Tri'Heem Neville

1	15	12
2	10	10
3	10	6
4	10	8.5
5	10	10
6	10+3	11
7	12	10 7
8	15	15
9	10	9
Total		88.5

Instructions:

Please keep your middle steps in the exam paper. Only the final result will NOT give you full credit.

You have 80 minutes to complete the exam. This exam counts for 25% of your course grade.

(15 pts) 1. Binary arithmetic and Boolean algebra

(3 pts) (1.1). Convert unsigned binary number to decimal number

$(110110.01)_2$

32	16	8	4	2	1	.5	25
1	1	0	1	1	0	.	1

$32 + 16 + 4 + 2 + .25$

$= 54.25$

(3 pts) (1.2). Convert the Hexadecimal number to binary number

B3F0

B = 1011

3 = 0011

F = 1111

0 = 0000

1011 0011 1111 0000

(3pts) (1.3). Please find the **complement** of the following function and simplify it.

$F = A'B(C+D') + E$

$F' = (A+B') + (C'D)E$

$= A + B' + C'DE$

(6 pts) (1.4). Please use the Boolean Algebra Theorems to simplify the following boolean expression to a minimum number of literals.

$F_1 = (BC' + A'D)(AB' + CD')$

$(BC'AB' + BC'CD') + (A'DAB' + A'DCD')$

$0 + 0 + 0 + 0$

$= 0$

$F_2 = (x(y'z' + yz))'$

$= x' + (y + z)(y' + z')$

$x' + (y\cancel{y'} + yz' + zy' + z\cancel{z})$

$= x' + (yz' + zy')$

(10 pts) 2. Decoder

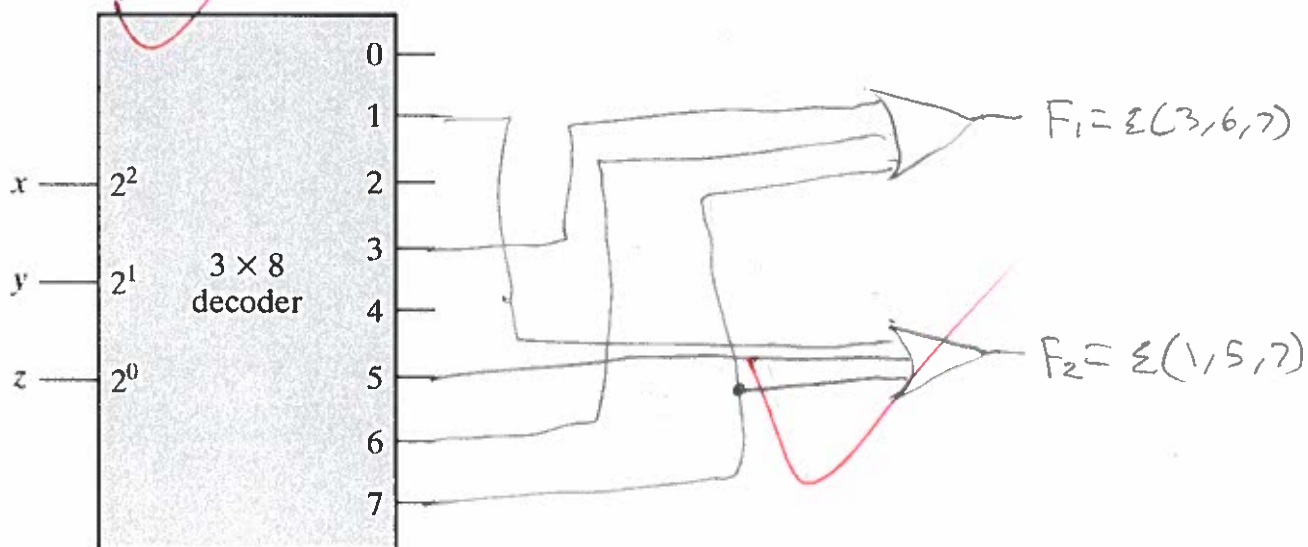
Use a decoder and external gate design the combination circuits (F_1 and F_2) defined by

$$F_1(x,y,z) = xy + yz$$

$$= \sum(3, 6, 7)$$

$$F_2(x,y,z) = xyz + y'z$$

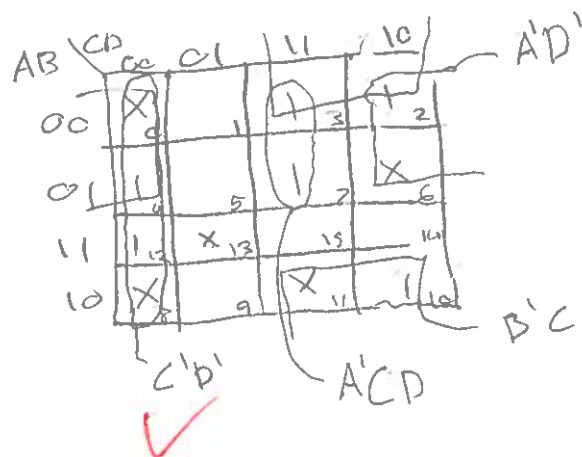
$$= \sum(1, 5, 7)$$



(10 pts) 3. Simplify the following Boolean Function F , together with the **Don't-Care** condition d , and then please express the function in **simplified Sum-of-Product** form:

$$F(A,B,C,D) = \sum_m(2, 3, 4, 7, 10, 12)$$

$$d(A,B,C,D) = \sum_m(0, 6, 8, 11, 13)$$



$$F = A'D' + B'C + C'D + A'CD$$

biggest!

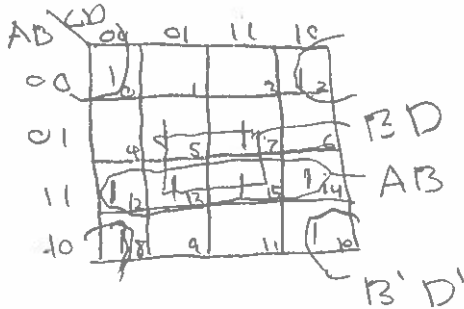
8.5

(10pts) 4. Prime implicants and Essential prime implicants

(a) Specify the following 4-variable boolean function in **simplified SOP form**.

(b) Please list the **Essential prime implicants**.

$$F(A,B,C,D) = \sum_m (0, 2, 5, 7, 8, 10, 12, 13, 14, 15)$$



a) $F = AB + BD + B'D'$
 b) EPI: $AB, BD, B'D'$

(10 pts) 5. Design a three-input, one-output minimal two-level gate combinatorial circuit using **only NAND** (you may use NOT gates if needed) gates that has a "1" output when only the majority of its inputs are "1", otherwise, output is "0".

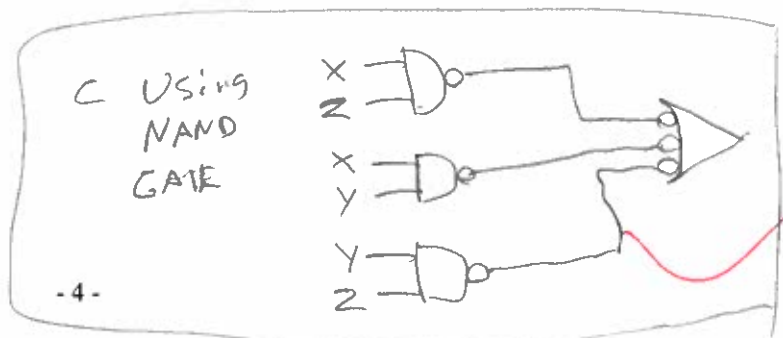
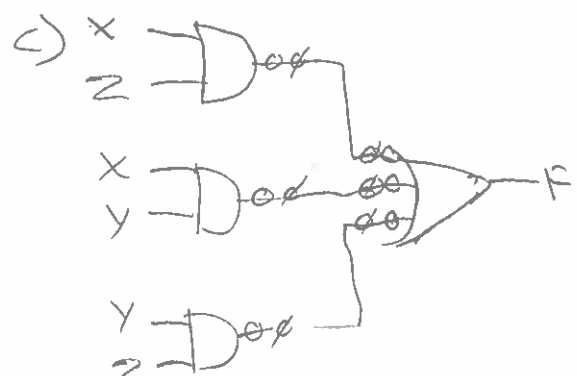
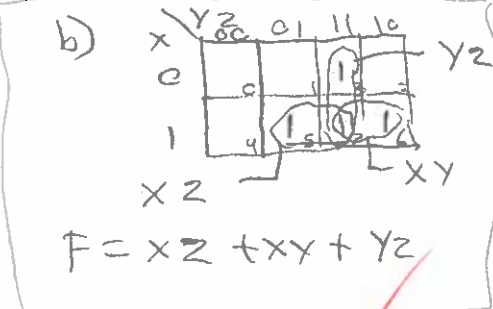
(3pt) a. Draw the truth table

(3pt) b. Minimize the function using K-Map

(4pt) c. Implement the logic using **only NAND** gates

a)

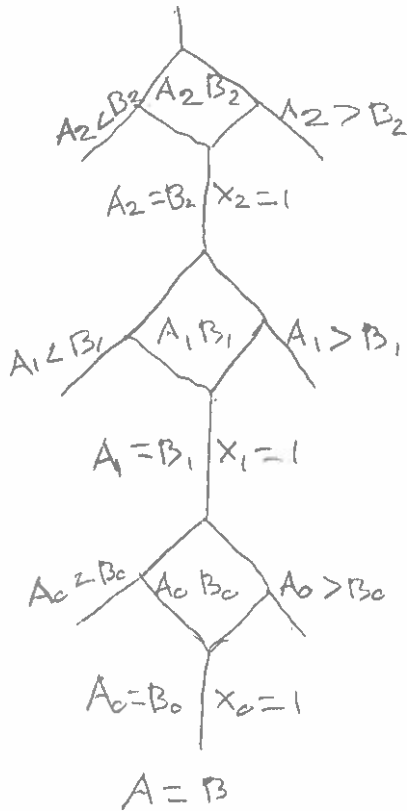
	X	Y	Z	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1



8+3

(10+3 pts) 6. Please design a logic to compare the magnitude of two 3-bit unsigned binary number (A, B): denoted by $A_3A_2A_1$ and $B_3B_2B_1$. Please show the boolean equations of how to determine whether one number is greater than ($A > B$), less than ($A < B$), or equal ($A = B$) to the other number.

Please show the design process, and demonstrate the boolean expressions and logic circuit (3 bonus points for correct logic circuit).

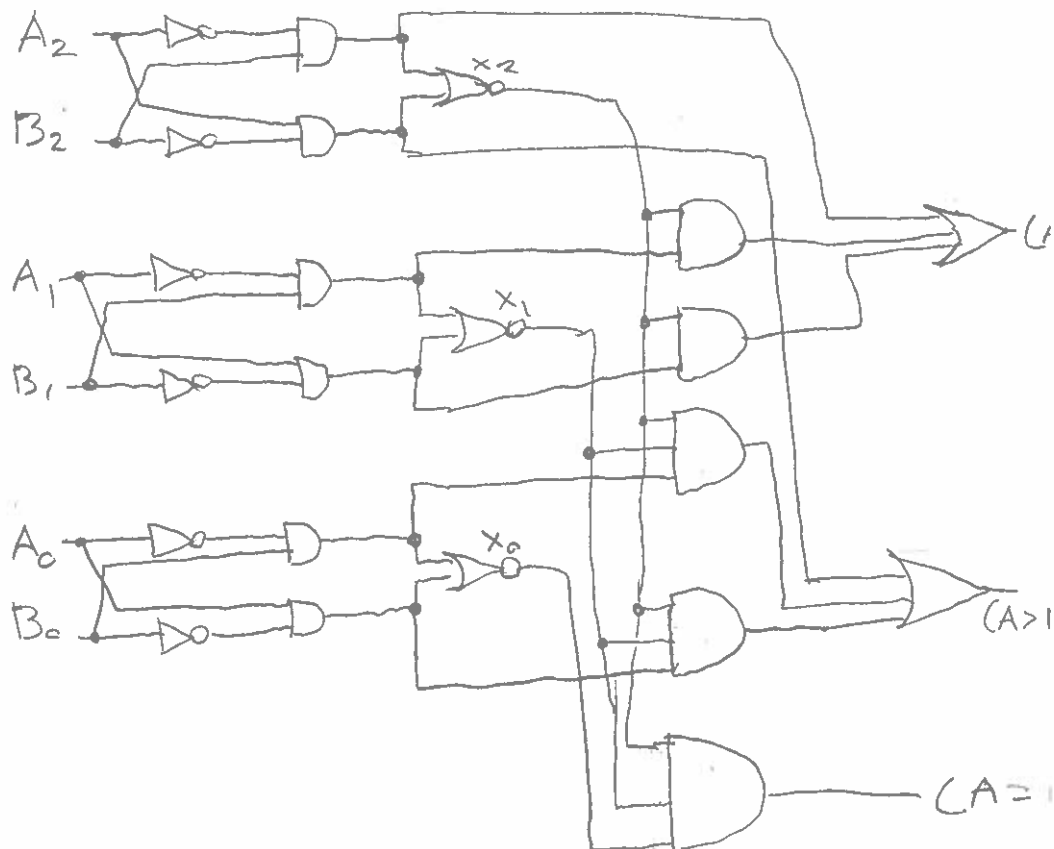


$$(A > B) = A_2 B_2' + x_2 A_1 B_1' + x_1 A_0 B_0'$$

$$(A < B) = A_2' B_2 + x_2 A_1' B_1 + x_1 A_0' B_0$$

$$(A = B) = x_2 x_1 x_0$$

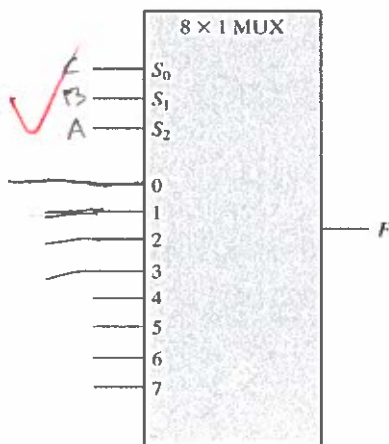
$$x_i = 0$$



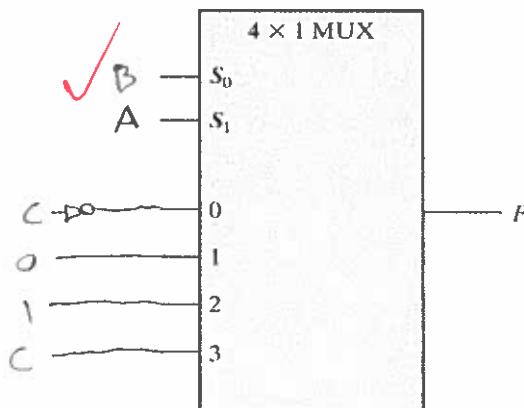
(12 pts) 7. Given the truth table below, please implement the truth table using a multiplexer.

(6 pts) (7.1) please using an 8-to-1 multiplexer to implement the truth table.

	S_2	S_1	S_0		
	A	B	C	F	
I_0	0	0	0	1	$F = 1$
	0	0	1	0	
	0	1	0	0	$F = 0$
I_1	0	1	1	0	
	1	0	0	1	$F = 1$
I_2	1	0	1	1	
	1	1	0	0	$F = 0$
I_3	1	1	1	1	$F = 1$



(6 pts) 7.2 please using a 4-to-1 multiplexer to implement the truth table. Briefly explain the design process.

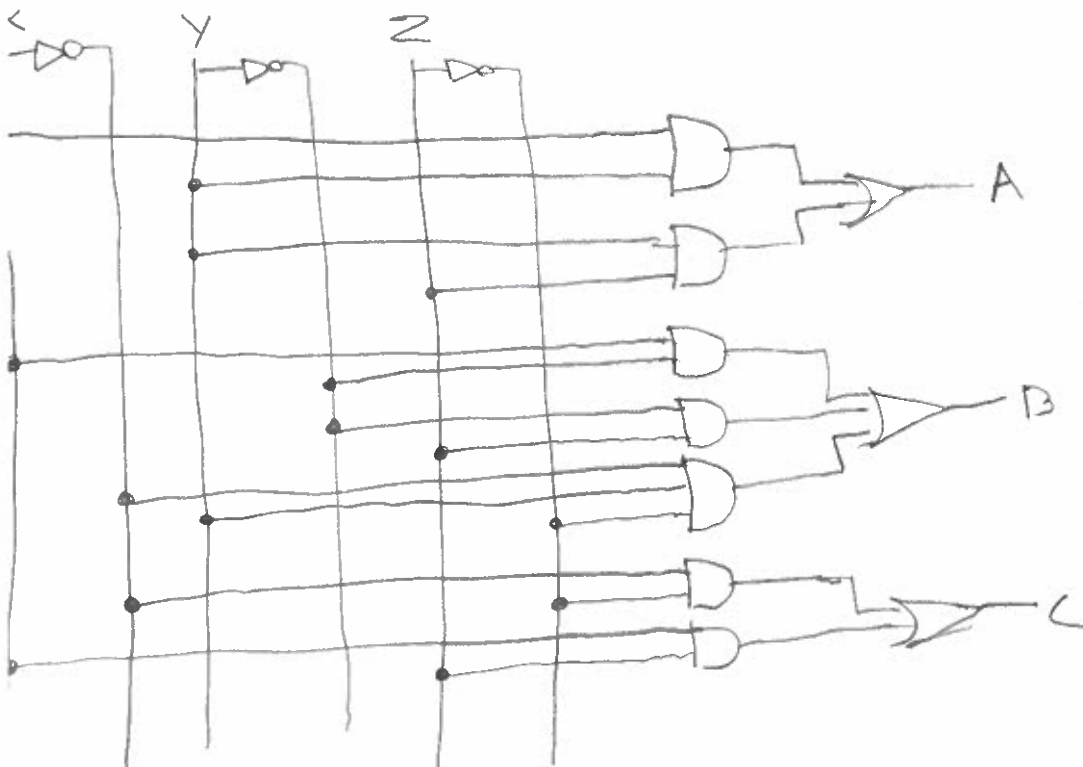
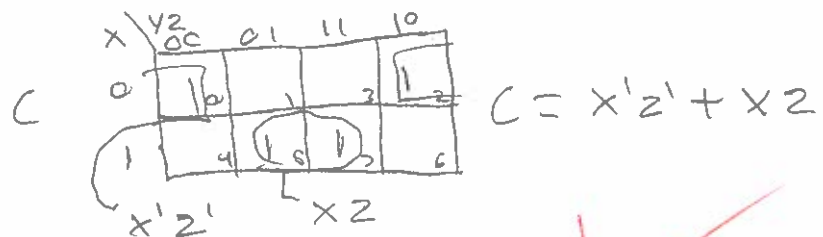
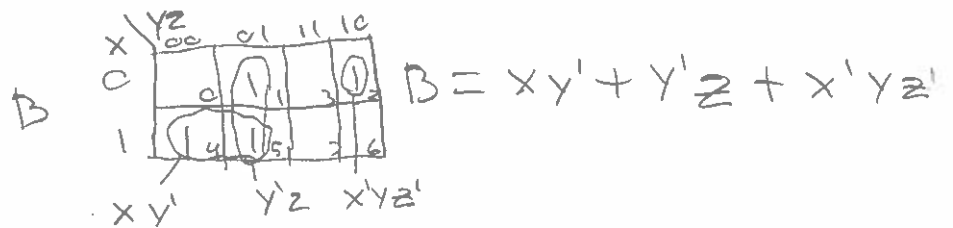
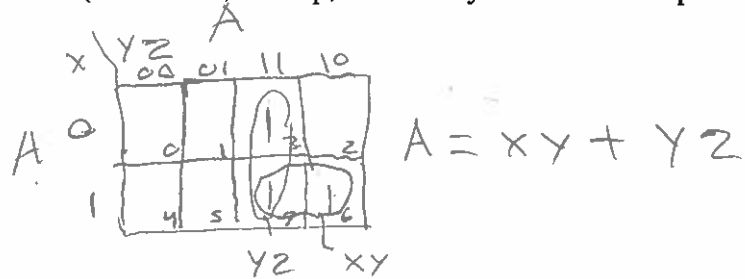


	$S_1 S_0$				
	A	B	C	F	
I_0	0	0	0	1	$F = C'$
	0	0	1	0	
I_1	0	1	0	0	$F = 0$
	0	1	1	0	
I_2	1	0	0	1	$F = 1$
	1	0	1	1	
I_3	1	1	0	0	
	1	1	1	1	$F = C$

- 15 (15 pts) 8. Design a combinational circuit with three inputs x, y, and z and three outputs A, B, and C. When the binary input is 0, 1, 2, or 3, the binary output is one greater than the input. When the binary input is 4, 5, 6, or 7, the binary output is two less than the input.

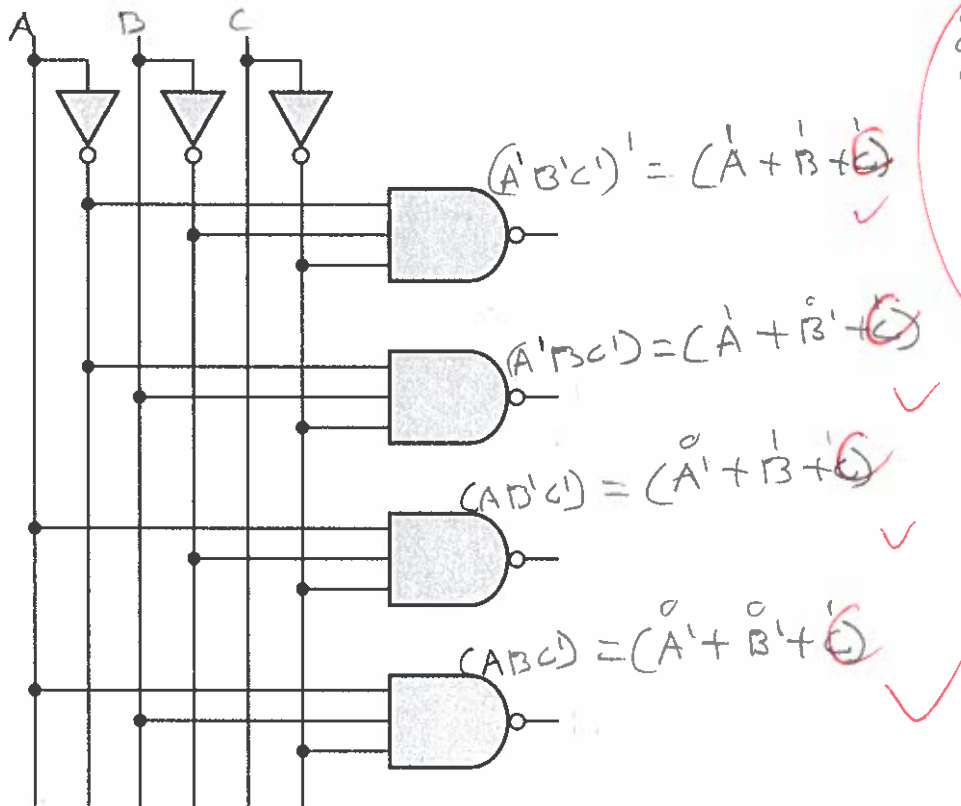
Please demonstrate the design process (truth table, K-map, and finally the boolean expression and logic circuits).

	X	Y	Z	A	B	C
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	0
4	1	0	0	0	1	0
5	1	0	1	0	1	1
6	1	1	0	1	0	0
7	1	1	1	1	0	1



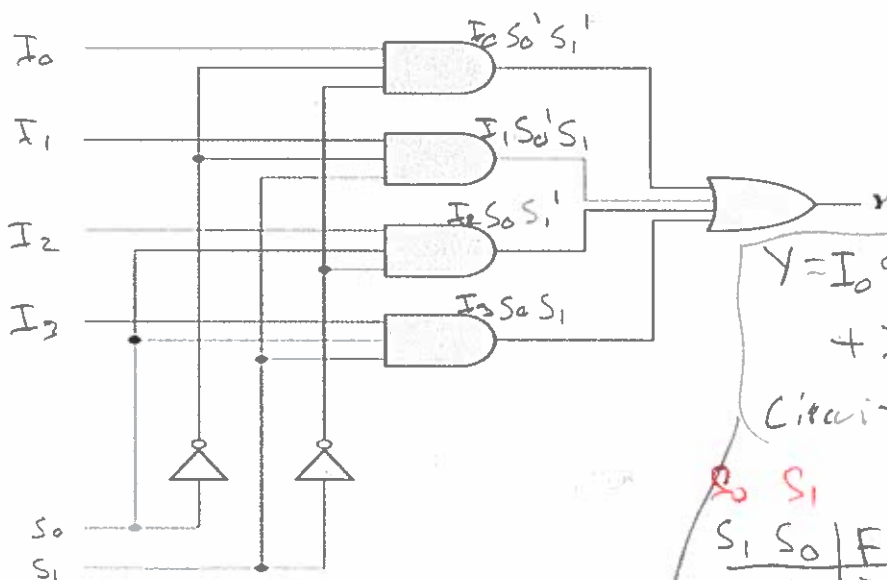
(10 pts) 9. Please analyze the logic, describe the function, and write the truth table/function table.

(5pt) 9.1



A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

(5pt) 9.2



$$Y = I_0 S_0 S_1' + I_1 S_0 S_1 + I_2 S_0 S_1' + I_3 S_0 S_1$$

Circuit is a 4 to 1 Mux

$S_1 S_0$

S_1	S_0	F
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Label

AVERAGE

ELEG 4330 Introduction to Digital Design
Department of Electrical & Computer Engineering
PVAMU

Midterm Exam

Fall 2021

Name: Javian Walker

1	15	15
2	10	10
3	10	10
4	10	8.5
5	10	10
6	10+3	8
7	12	9
8	15	0
9	10	1
Total		71.5

Instructions:

Please keep your middle steps in the exam paper. Only the final result will NOT give you full credit.

You have 80 minutes to complete the exam. This exam counts for 25% of your course grade.

(15 pts) 1. Binary arithmetic and Boolean algebra

(3 pts) (1.1). Convert unsigned binary number to decimal number

$$(110110.01)_2 = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$
$$= 32 + 16 + 4 + 2 + 0 + 0.25$$
$$= (54.25)_{10}$$

(3 pts) (1.2). Convert the Hexadecimal number to binary number

B3F0 $\Rightarrow$ $\left. \begin{array}{l} B = 1011 \\ 3 = 0011 \\ F = 1111 \\ 0 = 0000 \end{array} \right\} \therefore B3F0 = (1011\ 0011\ 1111\ 0000)_2$

(3pts) (1.3). Please find the **complement** of the following function and simplify it.

$$F = A'B(C+D') + E \Rightarrow F' = \overline{A'B(C+D') + E}$$
$$F' = \overline{A'B(C+D')} + \overline{E}$$
$$= (\overline{A'B} + \overline{(C+D')}) + E'$$
$$= ((A+B') + C'D) + E'$$
$$\therefore F' = AE' + B'E' + C'DE'$$

(6 pts) (1.4). Please use the Boolean Algebra Theorems to simplify the following boolean expression to a minimum number of literals.

$$F_1 = (BC' + A'D)(AB' + CD')$$
$$= BC'AB' + BC'CD' + A'DAB' + A'DCD'$$
$$* AA' = 0$$
$$= 0 + 0 + 0 + 0$$

$$\therefore F_1 = 0$$

$$F_2 = (x(y'z' + yz))'$$

$$= x' + (y'z' + yz)'$$

$$* (y'z' + yz)' = y \oplus z$$

$$= x' + (y \oplus z)$$

$$\therefore F_2 = x' + y'z + z'y$$

(10 pts) 2. Decoder

Use a decoder and external gate design the combination circuits (F_1 and F_2) defined by

$$F_1(x,y,z) = xy + yz$$

$$F_2(x,y,z) = xyz + y'z$$

$$F_1(x,y,z) = xy + yz$$

$$= xy(z + \bar{z}) + (x + \bar{x})yz$$

$$= xyz + xy\bar{z} + xyz + \bar{x}yz$$

$$= xyz + xy\bar{z} + \bar{x}yz$$

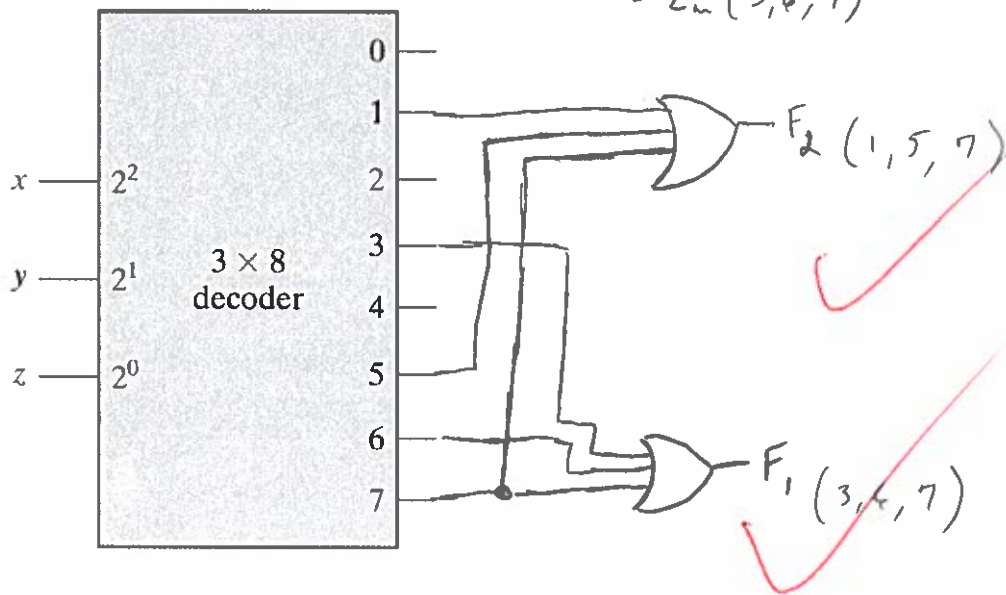
$$= \sum_m(3, 6, 7)$$

$$F_2(x,y,z) = xyz + \bar{y}z$$

$$= xyz + (x + \bar{x})\bar{y}z$$

$$= xyz + x\bar{y}z + \bar{x}\bar{y}z$$

$$= \sum_m(1, 5, 7)$$



10 (10 pts) 3. Simplify the following Boolean Function F , together with the Don't-Care condition d , and then please express the function in simplified Sum-of-Product form:

$$F(A,B,C,D) = \sum_m(2, 3, 4, 7, 10, 12)$$

$$d(A,B,C,D) = \sum_m(0, 6, 8, 11, 13)$$

AB \ CD	00	01	11	10
00	x		1	1
01	1		1	x
11	1	x		
10	x		x	1

$$\therefore F = \bar{C}\bar{D} + \bar{A}C + \bar{B}C$$

(10pts) 4. Prime implicants and Essential prime implicants

(a) Specify the following 4-variable boolean function in **simplified SOP form**.

(b) Please list the **Essential prime implicants**.

$$F(A,B,C,D) = \sum_m (0, 2, 5, 7, 8, 10, 12, 13, 14, 15)$$

AB \ CD	00	01	11	10
00	1			1
01		1	1	
11	1	1	1	1
10	1			1

$$\therefore F = \bar{B}\bar{D} + AB + BD$$

$$\therefore \text{EPI} = \bar{B}\bar{D} + \underline{AB} + BD$$

(10 pts) 5. Design a three-input, one-output minimal two-level gate combinatorial circuit using only NAND (you may use NOT gates if needed) gates that has a "1" output when only the majority of its inputs are "1", otherwise, output is "0".

(3pt) a. Draw the truth table

(3pt) b. Minimize the function using K-Map

(4pt) c. Implement the logic using **only NAND** gates

A.)

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

B.)

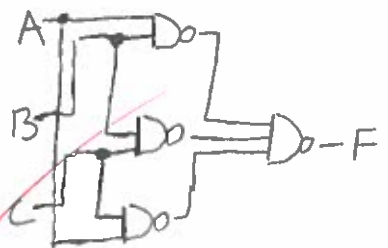
A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$F = BC + AB + AC$$

C) $F = BC + AB + AC$

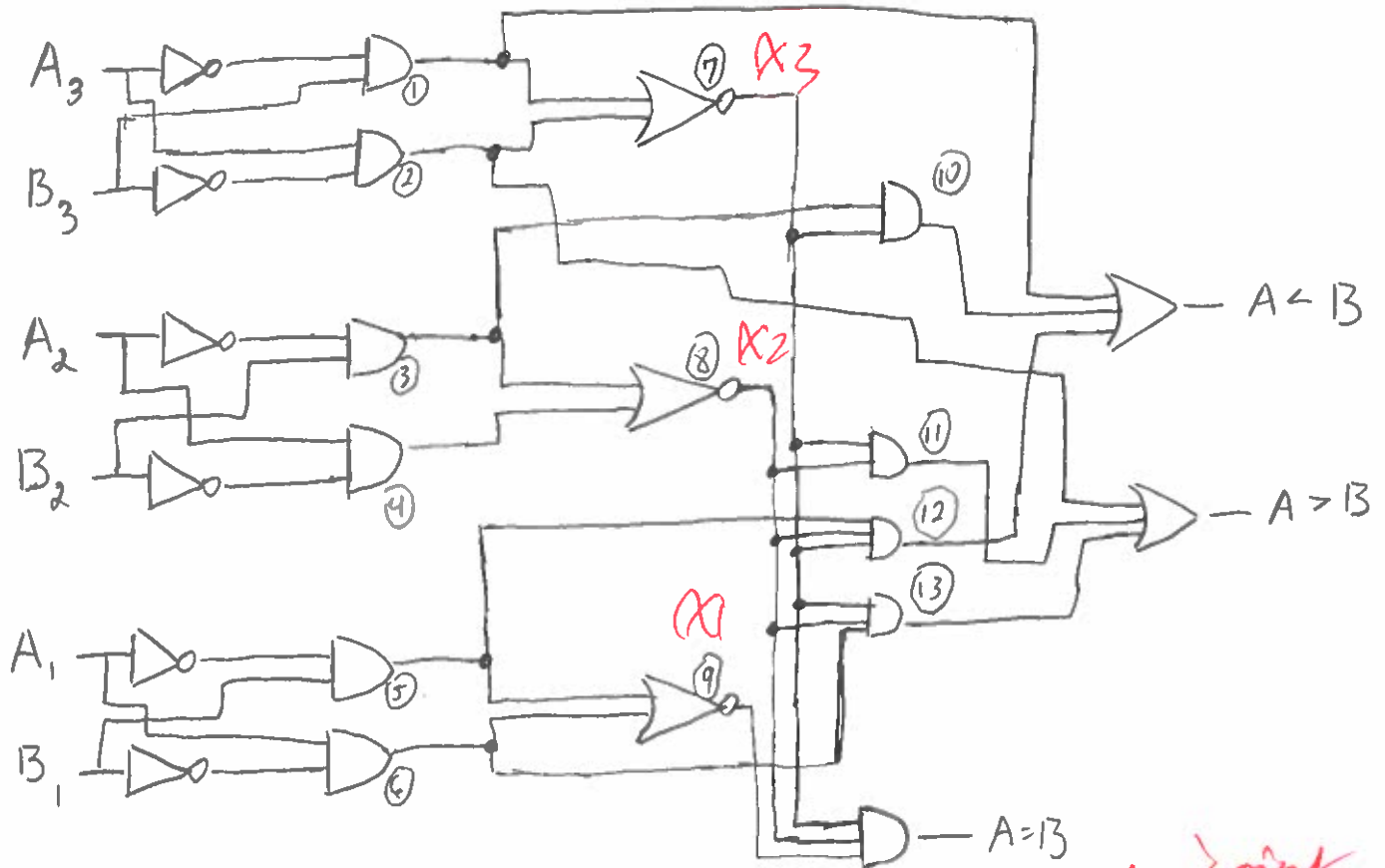
$$F = \overline{\overline{BC} + \overline{AB} + \overline{AC}}$$

$$F = \overline{\overline{BC}} \cdot \overline{\overline{AB}} \cdot \overline{\overline{AC}}$$



(10+3 pts) 6. Please design a logic to compare the magnitude of two 3-bit unsigned binary number (A, B): denoted by $A_3A_2A_1$ and $B_3B_2B_1$. Please show the boolean equations of how to determine whether one number is greater than ($A > B$), less than ($A < B$), or equal ($A = B$) to the other number.

Please **show the design process**, and demonstrate the **boolean expressions** and **logic circuit** (3 bonus points for correct logic circuit).



$$A < B = \bar{A}_3 B_3 + [(\bar{A}_2 B_2 (\bar{A}_3 B_3 + A_3 \bar{B}_3)')] \text{ when } \textcircled{1} + \textcircled{10} + \textcircled{12}$$

$$A > B = A_3 \bar{B}_3 + [(A_2 \bar{B}_2 (\bar{A}_3 B_3 + A_3 \bar{B}_3)')] + [A_2 \bar{B}_2 (\bar{A}_3 B_3 + A_3 \bar{B}_3)'] \text{ when } \textcircled{2} + \textcircled{11}$$

$$A = B \text{ when } \textcircled{7} + \textcircled{5} + \textcircled{9}$$

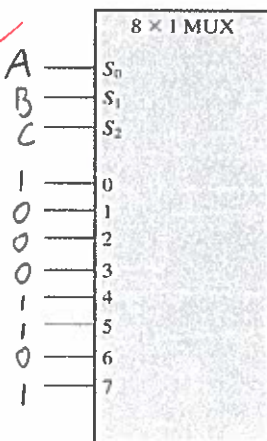
define $X_i = A_i B_i + A_i' B_i'$

(12 pts) 7. Given the truth table below, please implement the truth table using a multiplexer.

(6 pts) (7.1) please using an 8-to-1 multiplexer to implement the truth table.

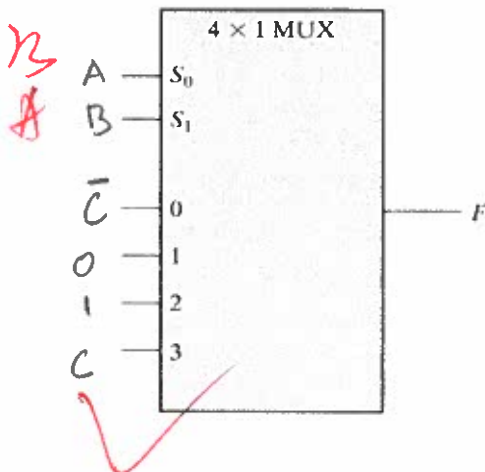
A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

A \ BC	00	01	11	10
00	$\bar{a}\bar{b}\bar{c}$	$\bar{a}\bar{b}c$	$\bar{a}b\bar{c}$	$\bar{a}bc$
01	$\bar{a}b\bar{c}$	$\bar{a}bc$	$a\bar{b}\bar{c}$	$a\bar{b}c$
11	$a\bar{b}\bar{c}$	$a\bar{b}c$	$ab\bar{c}$	abc
10	$ab\bar{c}$	abc	$ab\bar{c}$	abc



$$F = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC$$

(6 pts) 7.2 please using a 4-to-1 multiplexer to implement the truth table. Briefly explain the design process.



∴ To implement this 4x1 MUX based on the given Truth Table that has 3 inputs (A; B; C), you would input A into S_0 and B into S_1 . Then, to input C you would write as C or $\bar{C}$ for the 1 and 0 to make up for that 3rd column.

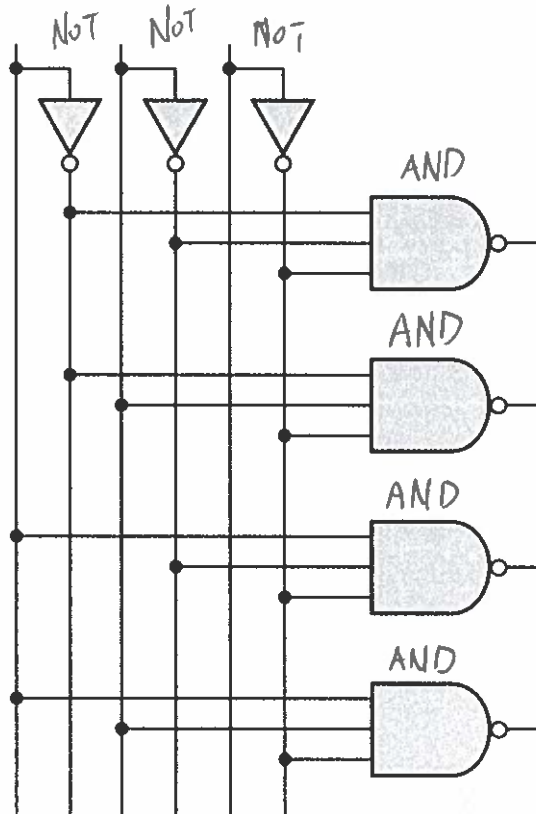
0 (15 pts) 8. Design a combinational circuit with three inputs x , y , and z and three outputs A , B , and C . When the binary input is 0, 1, 2, or 3, the binary output is one greater than the input. When the binary input is 4, 5, 6, or 7, the binary output is two less than the input.

Please demonstrate the design process (truth table, K-map, and finally the boolean expression and logic circuits).

(10 pts) 9. Please analyze the logic, describe the function, and write the truth table/function table.

(5pt) 9.1

NOT + AND = NAND



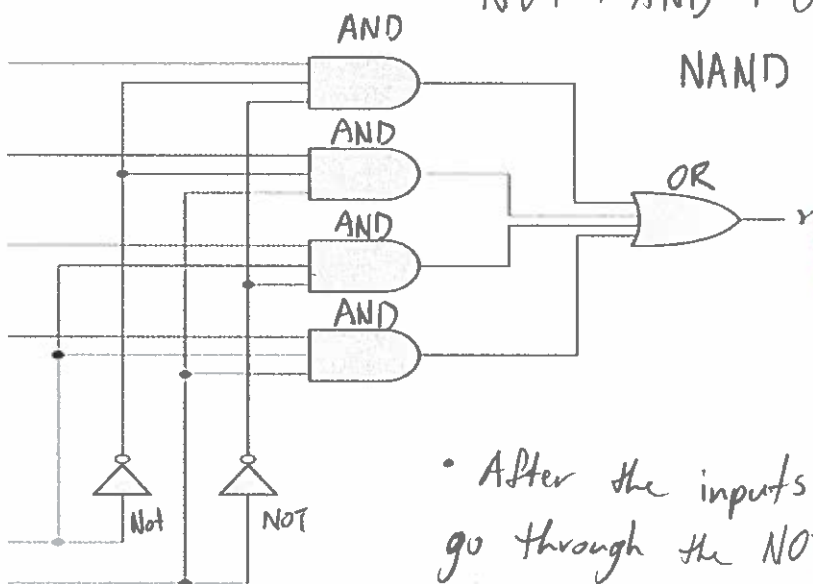
B	A	X
0	0	1
0	1	1
1	0	1
1	1	0

• Basically this logic is saying the inputs are split. 1 will go through the NOT and the other will go through the AND. So the output is basically "NOT x AND y".

(5pt) 9.2

NOT + AND + OR

NAND + OR = XNOR



B	A	X
0	0	1
0	1	0
1	0	0
1	1	1

• After the inputs go through the NOT gates and the AND gates the output would be "NAND x OR y".

BELOW AVERAGE

ELEG 4330 Introduction to Digital Design
Department of Electrical & Computer Engineering
PVAMU

Midterm Exam

Fall 2021

Name: Marvin Berrios

1	15	6.5
2	10	0
3	10	2
4	10	8
5	10	3
6	10+3	10
7	12	6
8	15	14.5
9	10	2.5
Total		52.5

Instructions:

Please keep your middle steps in the exam paper. Only the final result will NOT give you full credit.

You have 80 minutes to complete the exam. This exam counts for 25% of your course grade.

(15 pts) 1. Binary arithmetic and Boolean algebra

(3 pts) (1.1). Convert unsigned binary number to decimal number

$(110110.01)_2$ $2^6 + 2^5 + 2^3 + 2^2 + 2^{-2} = 108.25_{10}$

(3 pts) (1.2). Convert the Hexadecimal number to binary number

B3F0 $B=11_{10}$ $3=3_{10}$ $F=15_{10}$ $0=0_{10}$

$= (1011 \ 0011 \ 1111 \ 0000)_2$

(3pts) (1.3) . Please find the **complement** of the following function and simplify it.

$F = A'B(C+D') + E$

$F' = (A'B(C+D') + E)'$

$= (A'B(C+D'))' + E'$

$= A + B'(C'D) + E'$

(6 pts) (1.4). Please use the Boolean Algebra Theorems to simplify the following boolean expression to a minimum number of literals.

$F_1 = (BC' + A'D)(AB' + CD')$

$= AB'BC' + BC'CD' + AB'A'D + A'DCD'$

$= AC'(B'B) + BD'(C'C) + B'D(A'A) + A'C(D'D)$

$= AC' + BD' + B'D + A'C = (A \oplus C) + (B \oplus D)$

$F_2 = (x(y'z' + yz))'$

$= x' + (y+z)(y'+z')$

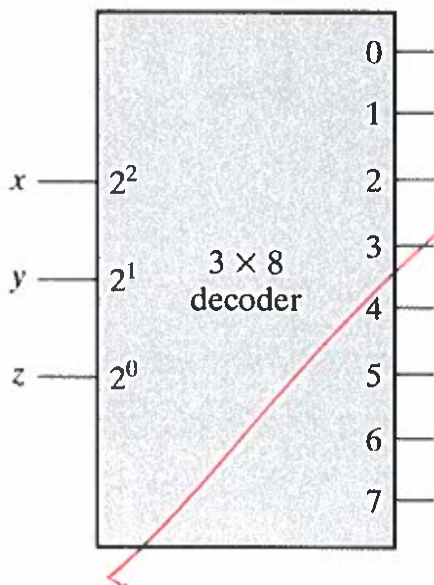
Simplify

(10 pts) 2. Decoder

Use a decoder and external gate design the combination circuits (F_1 and F_2) defined by

$$F_1(x,y,z)=xy+yz$$

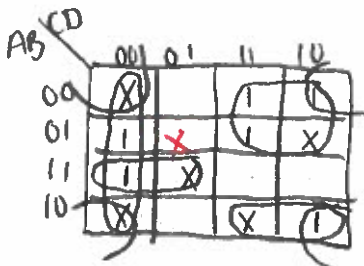
$$F_2(x,y,z)=xyz+y'z$$



(10 pts) 3. Simplify the following Boolean Function F , together with the **Don't-Care** condition d , and then please express the function in simplified Sum-of-Product form:

$$F(A,B,C,D)=\sum_m(2, 3, 4, 7, 10, 12)$$

$$d(A,B,C,D)=\sum_m(0, 6, 8, 11, 13)$$



$$F = C'D' + A'C + B'D' + ABC' + AB'C$$

$$= A'D'(C+C') + B'D' + ABC' + AB'C$$

$$= A'D' + B'D' + A(B+B') + A(C+C')$$

$$= A + A'D' + B'D'$$

Rules!

8 (10pts) 4. Prime implicants and Essential prime implicants

(a) Specify the following 4-variable boolean function in **simplified SOP form**.

(b) Please list the **Essential prime implicants**.

$$F(A,B,C,D) = \sum_m (0, 2, 5, 7, 8, 10, 12, 13, 14, 15)$$

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	1	1	1	1
10	1	0	0	1

$$F = B'D' + BD + AB$$

EPI

$B'D'$

0000	0101	1100
0010	0111	1110
1000	1011	1101
1010	1101	1111

3 (10 pts) 5. Design a three-input, one-output minimal two-level gate combinatorial circuit using **only NAND** (you may use NOT gates if needed) gates that has a "1" output when only the majority of its inputs are "1", otherwise, output is "0".

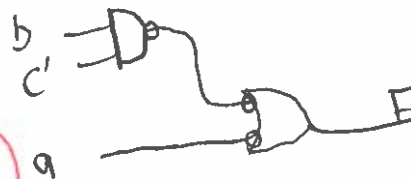
(3pt) a. Draw the truth table

(3pt) b. Minimize the function using K-Map

(4pt) c. Implement the logic using **only NAND** gates

a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$F = a' + bc'$$



a \ bc	00	01	11	10
0	0	1	1	0
1	0	0	0	1

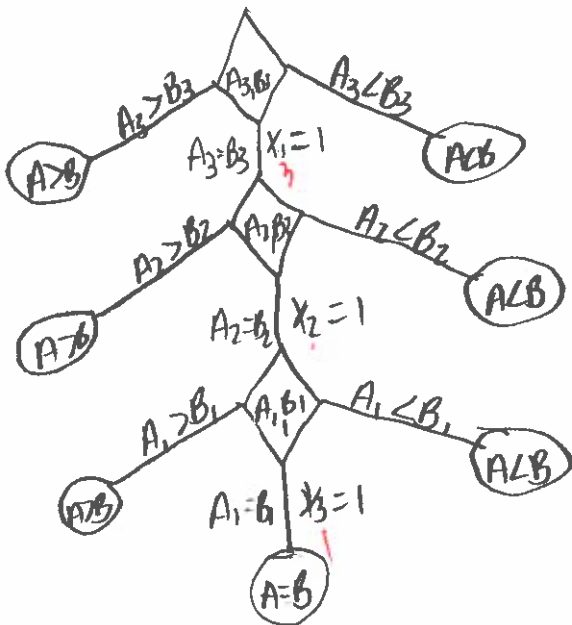
0000	0101	1100
0010	0111	1110
1000	1011	1101
1010	1101	1111

(10+3 pts) 6. Please design a logic to compare the magnitude of two 3-bit unsigned binary number (A, B): denoted by $A_3A_2A_1$ and $B_3B_2B_1$. Please show the boolean equations of how to determine whether one number is greater than ($A > B$), less than ($A < B$), or equal ($A = B$) to the other number.

Please show the design process, and demonstrate the boolean expressions and logic circuit (3 bonus points for correct logic circuit).



$$A = A_3A_2A_1 \quad B = B_3B_2B_1$$



$$X_i = A_i B_i + A_i' B_i'$$

$$\text{If } A_i = B_i$$

$$X_i = 1$$

$$(A = B) = X_3 \cdot X_2 \cdot X_1$$

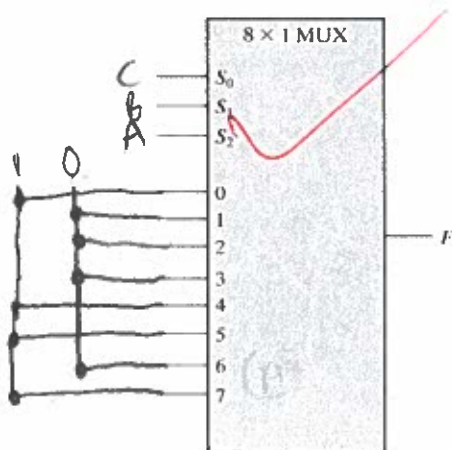
$$(A > B) = A_3 B_3' + A_2 B_2' X_3 + A_1 B_1' X_3 X_2$$

$$(A < B) = A_3' B_3 + A_2' B_2 X_3 + A_1' B_1 X_3 X_2$$

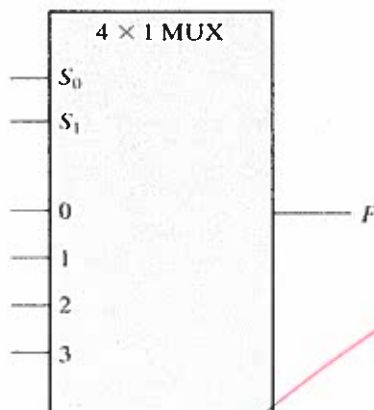
(12 pts) 7. Given the truth table below, please implement the truth table using a multiplexer.

(6 pts) (7.1) please using an 8-to-1 multiplexer to implement the truth table.

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



(6 pts) 7.2 please using a 4-to-1 multiplexer to implement the truth table. Briefly explain the design process.

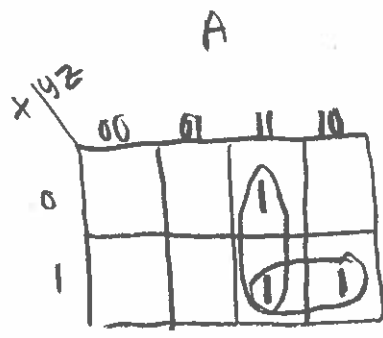


090 101

(15 pts) 8. Design a combinational circuit with three inputs x, y, and z and three outputs A, B, and C. When the binary input is 0, 1, 2, or 3, the binary output is one greater than the input. When the binary input is 4, 5, 6, or 7, the binary output is two less than the input.

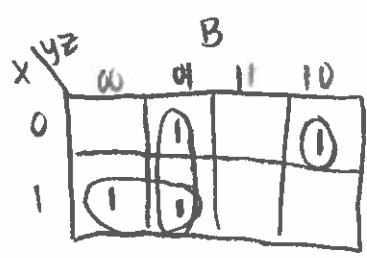
Please demonstrate the design process (truth table, K-map, and finally the boolean expression and logic circuits).

X	y	z	A	B	C
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	0	1

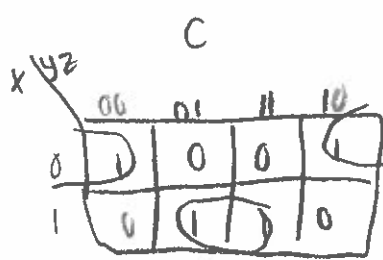


090 101

$$F = yz + xy$$

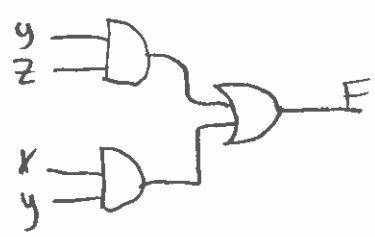


$$F = y'z + xy' + x'yz'$$

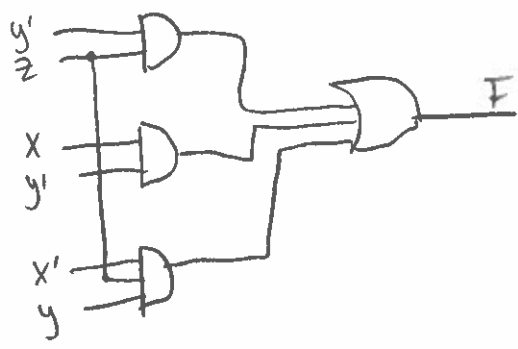


$$F = x'z' + xz$$

A



B



C

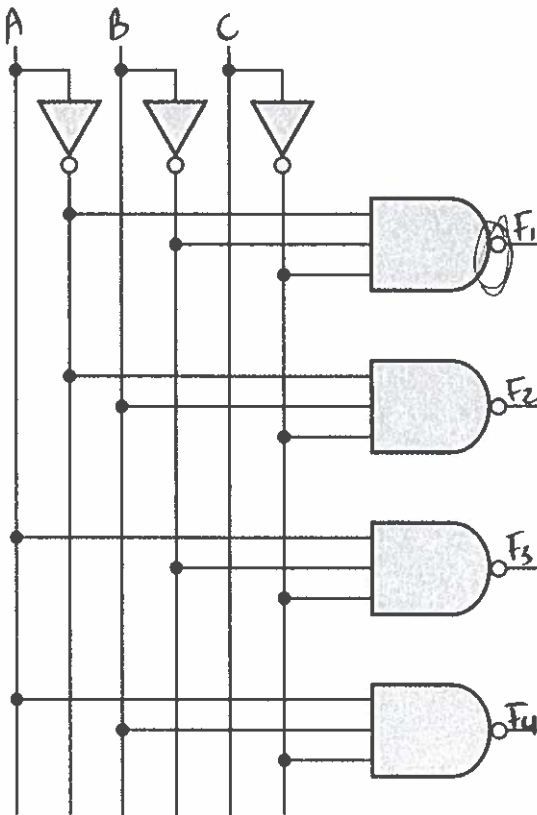


system view

010 100
110

(10 pts) 9. Please analyze the logic, describe the function, and write the truth table/function table.

(5pt) 9.1



A	B	C	F ₁	F ₂	F ₃	F ₄
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	1	0	0	1	0	0
0	1	1	0	0	0	0
1	0	0	0	0	1	0
1	0	1	0	0	0	0
1	1	0	0	0	0	1
1	1	1	0	0	0	0

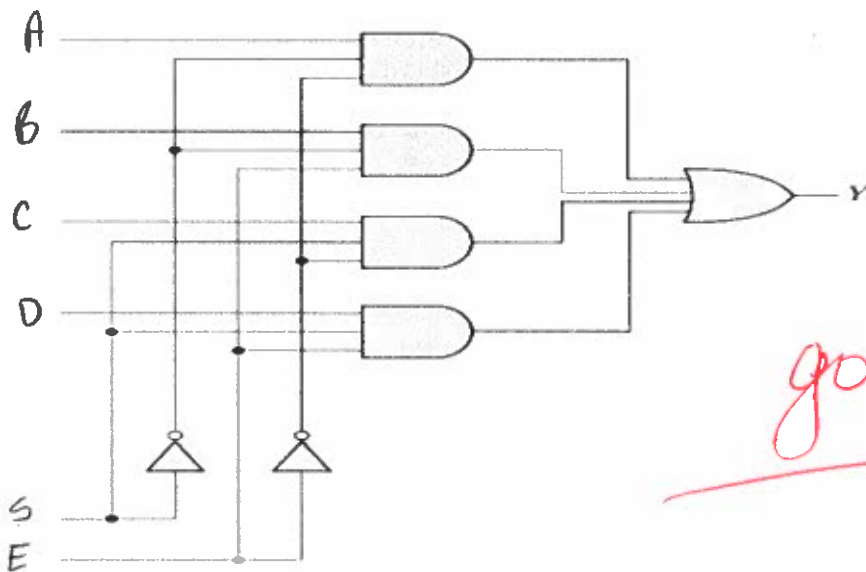
$$F_1 = A'B'C' \quad \times$$

$$F_2 = A'BC' \quad \times$$

$$F_3 = ABC' \quad \times$$

$$F_4 = ABC \quad \times$$

(5pt) 9.2



go on

ELEG 4330-P01 Introduction to Digital Design
Fall 2021

Title of Assignment: PROJECT 1

Combinational Logics Design: Code Converters

ELEG 4330-P01 Introduction to Digital Design
Fall 2021

Project 1

Electrical and Computer Engineering DEPARTMENT
OUTCOMES SPECIFIC ASSIGNMENT COVER SHEET
ELEG 4330 & Introduction to Digital Design
FALL 2021

Instructor: Dr. Xiangfang Li

Title of Assignment

Combinational Logics Design: Code Converters

Outcome Measured Using this Assignment

Outcome 2

An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factor

Check the engineering design characteristics related to this assignment

X	Involves identifying the specific needs to be met by defining/understanding the problem and developing plans for the project to satisfy the need
X	Involves conducting a review of the literature and performing preliminary design by generating alternative ideas/solutions and applying creativity
X	Involves performing detailed engineering design leading to specification and fabrication/selection of components for system being designed.
	Involves consideration of one or more of the following (Please check those involved in this assignment.)
	Public health
	Safety and welfare
	Global factors
	Cultural factors
	Social factors
	Environmental factors
x	Economic factors
	Involves fabrication and testing of model or prototype

Brief justification of the suitability of assignment for the outcome

--

Prairie View A&M University
Department of Electrical and Computer Engineering
ELEG 4330 & Introduction to Digital Design

FALL 2021 (Due Nov. 9)

Students should upload this project report to TaskStream Assignment and eCourses.

Project 1: Combinational Logics Design: Code Converters

The conversion from one binary code to another is common in digital systems. In this project, the conversion between binary code and Gray code will be designed and implemented. A project report is required.

Project Aim: To realize Binary code to Gray code converter and vice versa.

Learning Objective:

1. To learn the importance of non-weighted binary code.
2. To get familiar with combinational circuits design procedure.
3. Understand the importance of Universal gates.

Project requirements:

Design a combinational circuit with four inputs and four outputs:

- (1) The circuit should convert a four-bit Gray code number (textbook table 1.6) into the equivalent four-bit binary number. Please implement the circuit with Universal gate (NAND gates, in this case).
- (2) The logic circuit could convert a four-bit binary number into the equivalent four-bit Gray code. Please implement the circuit with exclusive-OR gate.

Procedure:

- (1) Please demonstrate the design process, from truth table, K-map, corresponding Boolean expressions, and finally logic circuits implementation.
- (2) Please discuss the following questions in your report. (a) What is the necessity of code conversions? (b) What is gray code? (c) What are universal gates? Why it is important?
- (3) If you could access the lab, you could physically implement the logic using appropriate ICs. If you could not access the lab, you could use Multisim to simulate the logic circuits.
- (4) Please test whether your design is correct, and demonstrate the testing procedure in your report.

In your report, please demonstrate your design process to produce the needed results based on the knowledge you learned in this course, with the consideration of how to make the design more optimal, more efficient, and at the same time with economical consideration and limitation.

Rubrics for Grading Design Project

ITEMS	POINTS ACHIEVED	MAXIMUM POINTS
1. Introduction		10
2. Demonstrate design process, including truth table, K-Map, Boolean expressions, etc		30
3. Your final logic circuits and simulation (using Multisim or ICs); Analysis and Interpretation of your design		30
4. Discussion of Results and Conclusions		20
5. General Write-up of report • Grammar and adequacy of plots and figures • Proper labeling and reference of graphs and tables • Clear and meaningful sentences		10
6. Composite Report Grade		100
7. Grade for student Outcome 2 Outcome 2: An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factor		100

ABOVE AVERAGE

95



Course Number: ELEG 4330

Combinational Logics Design: Code Converters

Introduction to Digital Design

Submitted by:

Mohamed Ahmed

Date: 11/09/2021

Professor: Dr. Xiangfang

Table of Contents

1. Introduction	3
2. Background.....	3
3. Gray Code to Binary Converter Methodology	3
3.1 Gray Code to Binary - Truth Table	4
3.2 Gray Code to Binary - K-Maps	4
3.3 Gray Code to Binary Converter - Logic circuit implementation	6
3.4 Testing Gray code to Binary Converter	7
4. Binary to Gray Code Converter Methodology	15
4.1 Binary to Gray code - Truth Table.....	15
4.2 Binary to Gray code K-Maps.....	16
4.3 Binary to Gray Code converter - Logic Circuit Implementation	17
4.4 Testing Binary to Gray code converter.....	18
5. Results	26
6. Conclusion.....	26
References.....	27

Figure 1: K- Map for Gray Code to Binary converter (Output W).....	4
Figure 2: K- Map for Gray Code to Binary converter (Output X).....	5
Figure 3: K- Map for Gray Code to Binary converter (Output Y).....	5
Figure 4: K- Map for Gray Code to Binary converter (Output Z).....	5
Figure 5: Logic Circuit Implementation for Gray Code to Binary converter.....	6
Figure 6: Testing Gray Code to Binary Converter (Gray Code value 0000).....	7
Figure 7: Testing Gray Code to Binary Converter (Gray Code value 0001).....	7
Figure 8: Testing Gray Code to Binary Converter (Gray Code value 0011).....	8
Figure 9: Testing Gray Code to Binary Converter (Gray Code value 0010).....	8
Figure 10: Testing Gray Code to Binary Converter (Gray Code value 0110).....	9
Figure 11: Testing Gray Code to Binary Converter (Gray Code value 0111).....	9
Figure 12: Testing Gray Code to Binary Converter (Gray Code value 0101).....	10
Figure 13: Testing Gray Code to Binary Converter (Gray Code value 0100).....	10
Figure 14: Testing Gray Code to Binary Converter (Gray Code value 1100).....	11
Figure 15: Testing Gray Code to Binary Converter (Gray Code value 1101).....	11
Figure 16: Testing Gray Code to Binary Converter (Gray Code value 1111).....	12
Figure 17: Testing Gray Code to Binary Converter (Gray Code value 1110).....	12
Figure 18: Testing Gray Code to Binary Converter (Gray Code value 1010).....	13
Figure 19: Testing Gray Code to Binary Converter (Gray Code value 1011).....	13
Figure 20:- Testing Gray Code to Binary Converter (Gray Code value 1001).....	14
Figure 21: Testing Gray Code to Binary Converter (Gray Code value 1000).....	14
Figure 22: K- Map for Binary to Gray Code converter (Output W).....	16
Figure 23: K- Map for Gray Code to Binary converter (Output X).....	16
Figure 24: K- Map for Binary to Gray Code converter (Output Y).....	16
Figure 25: K- Map for Binary to Gray Code converter (Output Z).....	17
Figure 26: Binary to Gray Code converter - Logic Circuit Implementation.....	17
Figure 27: Testing Binary to Gray Code Converter (Binary value 0000).....	18
Figure 28; Testing Binary to Gray Code Converter (Binary value 0001).....	18
Figure 29: Testing Binary to Gray Code Converter (Binary value 0010).....	19
Figure 30: Testing Binary to Gray Code Converter (Binary value 0011).....	19
Figure 31: Testing Binary to Gray Code Converter (Binary value 0100).....	20
Figure 32: Testing Binary to Gray Code Converter (Binary value 0101).....	20
Figure 33: Testing Binary to Gray Code Converter (Binary value 0110).....	21
Figure 34: Testing Binary to Gray Code Converter (Binary value 0111).....	21
Figure 35: Testing Binary to Gray Code Converter (Binary value 1000).....	22
Figure 36: Testing Binary to Gray Code Converter (Binary value 1001).....	22
Figure 37: Testing Binary to Gray Code Converter (Binary value 1010).....	23
Figure 38: Testing Binary to Gray Code Converter (Binary value 1011).....	23
Figure 39: Testing Binary to Gray Code Converter (Binary value 1100).....	24
Figure 40: Testing Binary to Gray Code Converter (Binary value 1101).....	24
Figure 41: Testing Binary to Gray Code Converter (Binary value 1110).....	25
Figure 42:Testing Binary to Gray Code Converter (Binary value 1111).....	25

1. Introduction:

The conversion from one binary code to another is commonplace in digital systems and is an essential tool. In this project, two code converters were designed. The first code converter takes a four-bit Gray code number and converts it to its equivalent four-bit binary number using NAND gates. The second code converter designed converts a four-bit binary number into the equivalent four-bit Gray code using exclusive-OR gates. This project explores the importance of non-weighted binary code, the combinational circuit design procedures, and the significance of universal gates.

2. Background

In weighted codes, each digit is assigned a fixed weight according to its position. For example, the BCD code has weights of 8, 4, 2, and 1, which match each bit's power-of-two values. BCD obeys the positional weight principle. In a non-weighted binary code, each position is not assigned a fixed value. Gray code is an example of a non-weighted code. Gray code is used in applications where the normal sequence of binary numbers generated by the hardware may produce an error during the transition from one number to the next. Numbers where all the bits flip, for example, 011 to 100, may have an incorrect number. Gray code addresses this problem since only one bit changes its value during the transition. It may be more beneficial to use codes like binary, octal, hexadecimal, Binary Coded Decimal (BCD), Excess-3, or Gray code in different situations.

A combinational circuit comprises interconnections of logic gates. Combinational logic gates react to the value of the signals at their input and generate the output signal value. The output in a combinational circuit is a function of the present input only. A universal gate is a gate that can implement any Boolean function without the necessity of any other gate type. The NAND and exclusive-OR gates are examples of universal gates. Both the NAND and exclusive-OR gates have tremendous advantages for their economical and ease of fabrication.

3. Gray Code to Binary Converter Methodology

Design procedures to convert a four-bit Gray code number to its equivalent four-bit binary number using NAND gates required the use of the truth table, K-Maps, Boolean expressions, and Multisim to implement the logic circuit. Binary and Gray code uses four bits; the converter must have four input and output variables. The four input Gray code variables are represented by the symbols A, B, C, and D and the four output variables by W, X, Y, and Z. The truth table relating the input and the output is illustrated in Table 1. The K-maps (Fig. 1 through 4) are plotted to obtain simplified Boolean functions for the outputs. Each of the four maps represents one of the four outputs of the circuit as a function of the four input variables. From the K-maps, the Boolean expression is obtained. By manipulating the Boolean expression, a logic diagram using NAND gates was created. The Boolean expressions were then used to implement the circuit via Multisim. Testing was conducted by switching the switches (ABCD) to on (1) or off (0) to correspond to its Gray code value from the truth table. The binary values produced by the circuit

were then compared to the truth table (table 1). All Gray code (0-15) values were checked and produced the correct values.

3.1 Gray Code to Binary - Truth Table

Decimal Number	Gray Code ABCD	Binary WXYZ
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111

Table 1- Truth Table for Gray Code to Binary converter

3.2 Gray Code to Binary - K-Maps

$W = A$

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

Figure 1- K- Map for Gray Code to Binary converter (Output W)

$$X = A \oplus B$$

$$X = A'B + AB'$$

$$X = ((A'B) + (AB'))'$$

$$X = (A'B)'(AB')'$$

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

Figure 2: K- Map for Gray Code to Binary converter (Output X)

$$Y = A \oplus B \oplus C$$

$$Y = (A'B'C) + (A'BC') + (ABC) + (AB'C')$$

$$Y = ((A'B'C) + (A'BC') + (ABC) + (AB'C'))'$$

$$Y = ((A'B'C)'(A'BC')'(ABC)'(AB'C'))'$$

AB \ CD	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

Figure 3: K- Map for Gray Code to Binary converter (Output Y)

$$Z = A \oplus B \oplus C \oplus D$$

$$= A'B'C'D + A'B'CD' + A'BC'D' + A'BCD + ABC'D + ABCD' + AB'C'D' + AB'CD$$

$$= ((A'B'C'D) + (A'B'CD') + (A'BC'D') + (A'BCD) + (ABC'D) + (ABCD') + (AB'C'D') + (AB'CD))'$$

$$Z = ((A'B'C'D)'(A'B'CD')'(A'BC'D')'(A'BCD)'(ABC'D)'(ABCD')'(AB'C'D')'(AB'CD))'$$

AB \ CD	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

Figure 4: K- Map for Gray Code to Binary converter (Output Z)

Output Boolean expression:

$$W = A$$

$$X = (A'B)'(AB')'$$

$$Y = ((A'B'C)'(A'BC')'(ABC)'(AB'C'))'$$

$$Z = ((A'B'C'D)'(A'B'CD')'(A'BC'D')'(A'BCD)'(ABC'D)'(ABCD')'(AB'C'D')'(AB'CD))'$$

3.3 Gray Code to Binary Converter - Logic circuit implementation

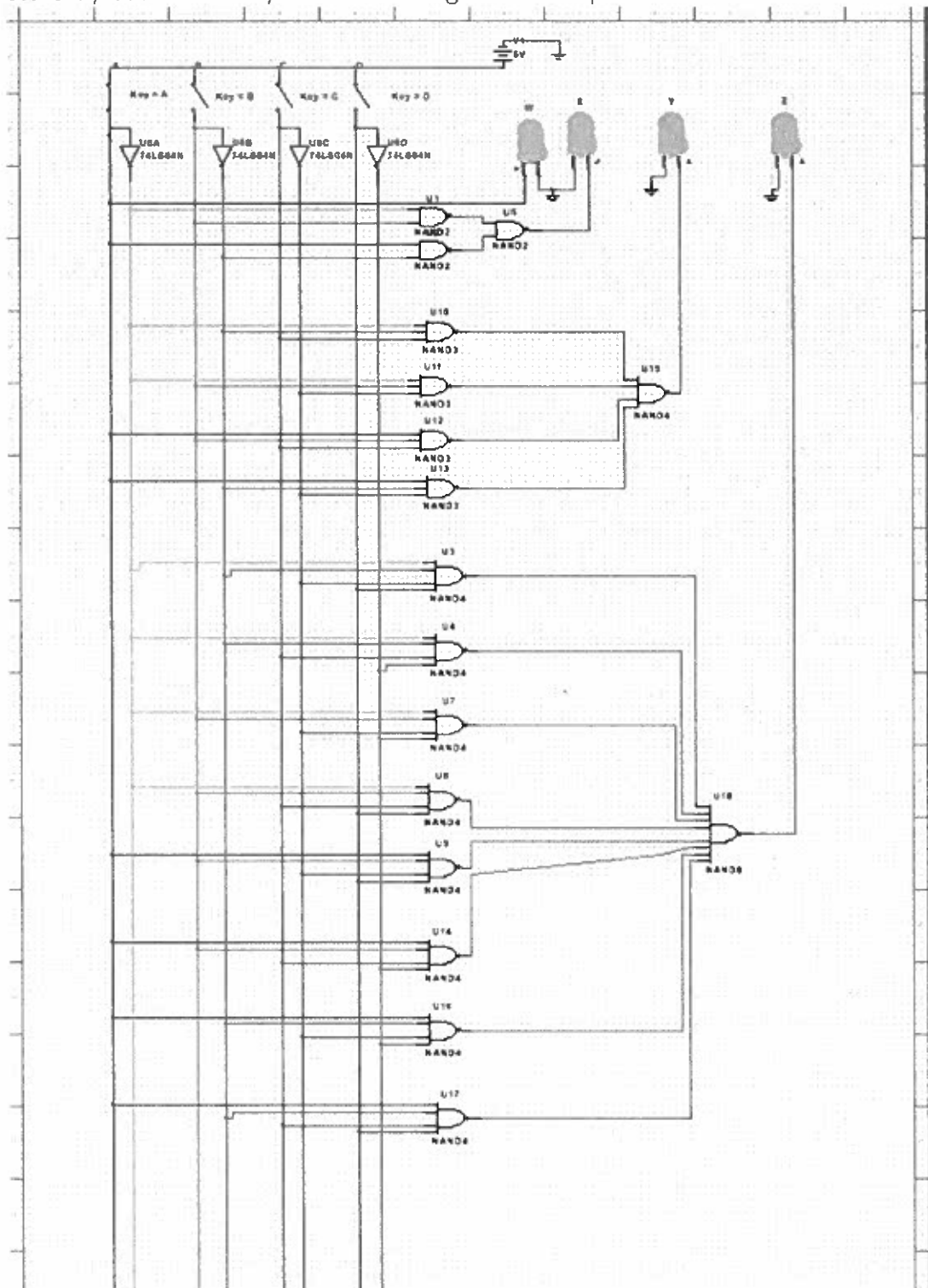


Figure 5: Logic Circuit Implementation for Gray Code to Binary converter

3.4 Testing Gray code to Binary Converter

Testing was conducted by switching the switches (ABCD) to on (1) or off (0) to correspond to its Gray code values from the truth table. The Binary values (WXYZ) was linked to a LED. LED that was on communicated a value of 1, while if the LED remained off, the value expressed equaled 0. The logic circuit's binary values (WXYZ) were then compared to the truth table (table 1). All Gray code values correctly translated to their Binary value.

Decimal Number	Gray Code ABCD	Binary WXYZ
0	0000	0000

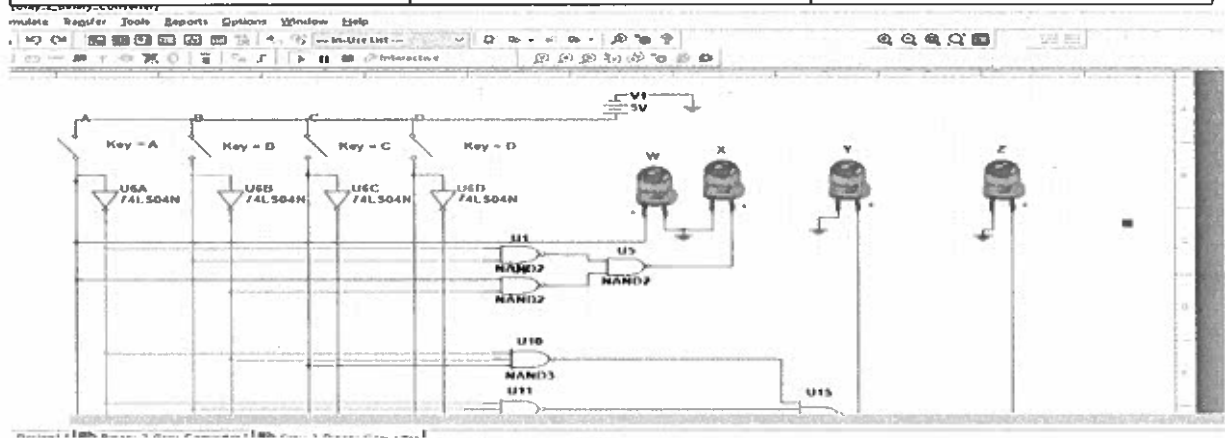


Figure 6: Testing Gray Code to Binary Converter (Gray Code value 0000)

Decimal Number	Gray Code ABCD	Binary WXYZ
1	0001	0001

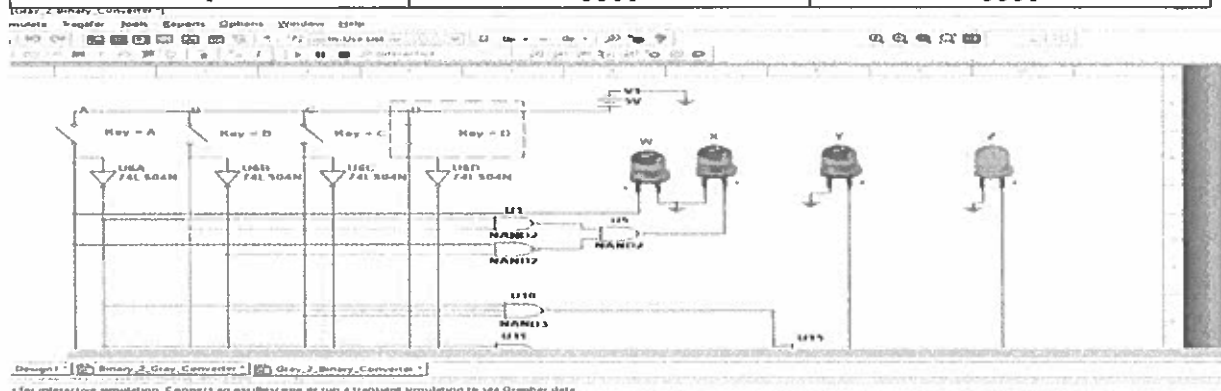


Figure 7: Testing Gray Code to Binary Converter (Gray Code value 0001)

Decimal Number	Gray Code ABCD	Binary WXYZ
2	0011	0010

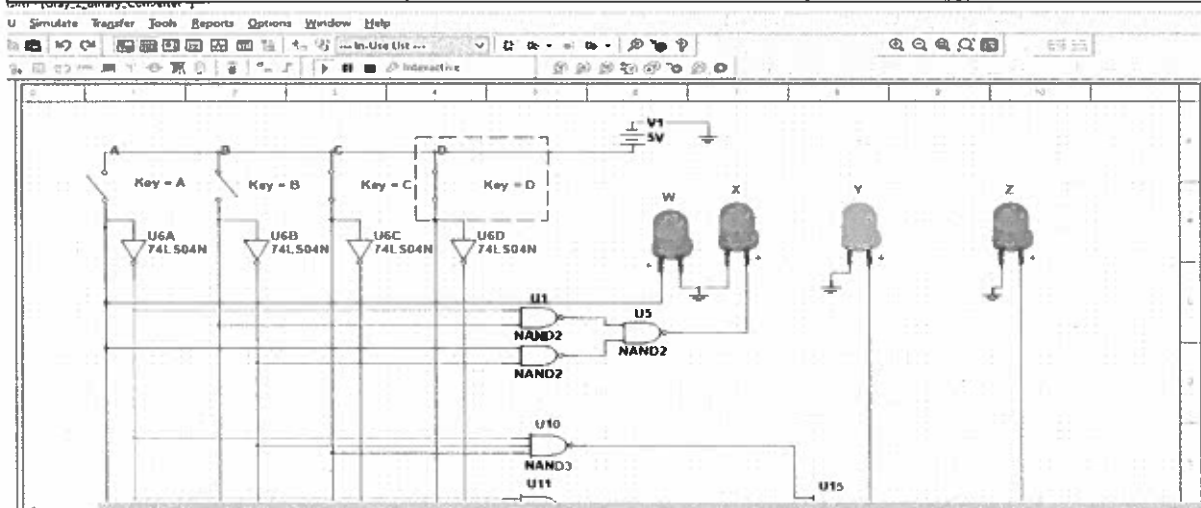


Figure 8. Testing Gray Code to Binary Converter (Gray Code value 0011)

Decimal Number	Gray Code ABCD	Binary WXYZ
3	0010	0011

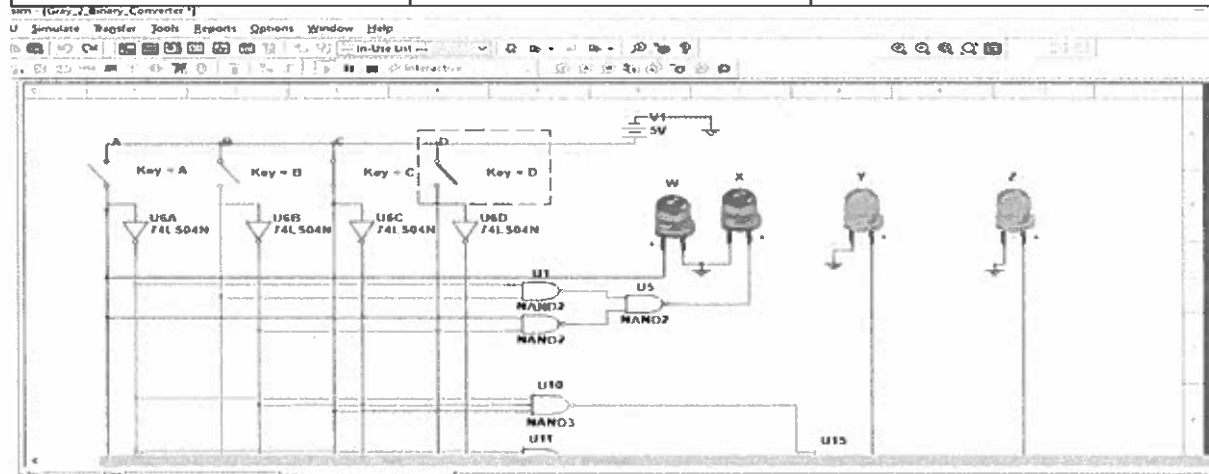


Figure 9. Testing Gray Code to Binary Converter (Gray Code value 0010)

Decimal Number	Gray Code ABCD	Binary WXYZ
6	0101	0110

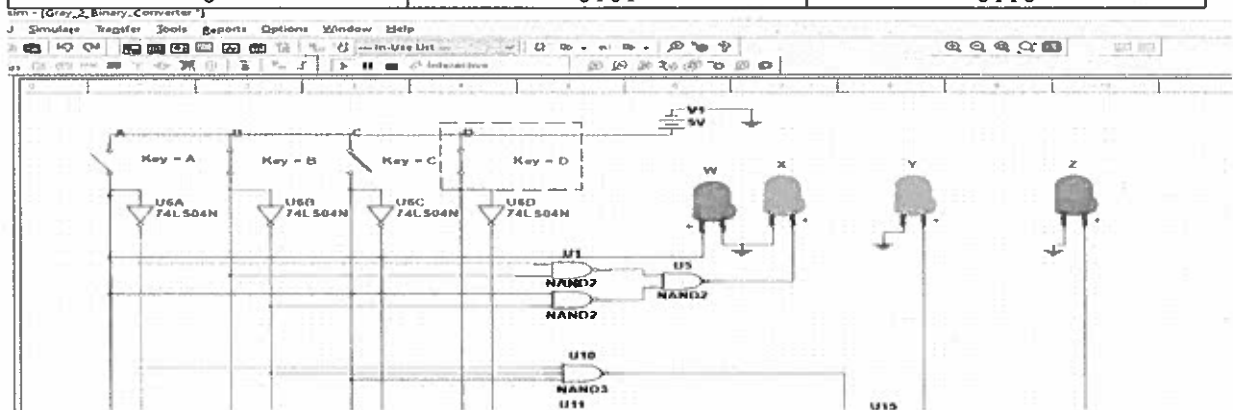


Figure 12: Testing Gray Code to Binary Converter (Gray Code value 0101)

Decimal Number	Gray Code ABCD	Binary WXYZ
7	0100	0111

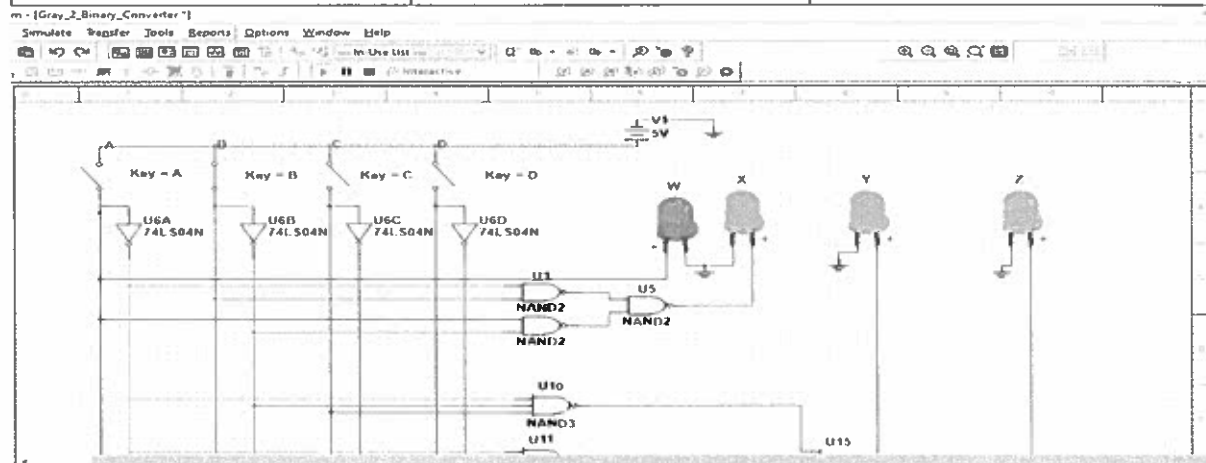


Figure 13: Testing Gray Code to Binary Converter (Gray Code value 0100)

Decimal Number	Gray Code ABCD	Binary WXYZ
8	1100	1000

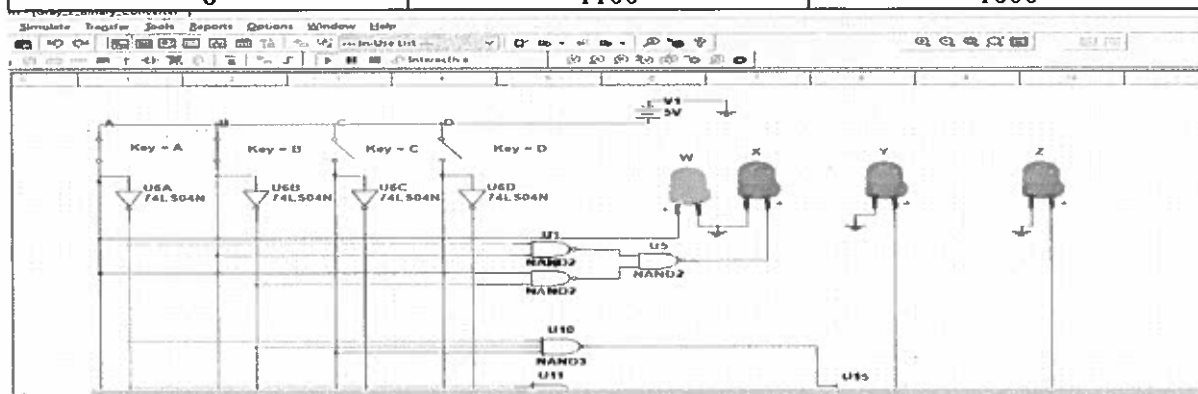


Figure 14: Testing Gray Code to Binary Converter (Gray Code value 1100)

Decimal Number	Gray Code ABCD	Binary WXYZ
9	1101	1001

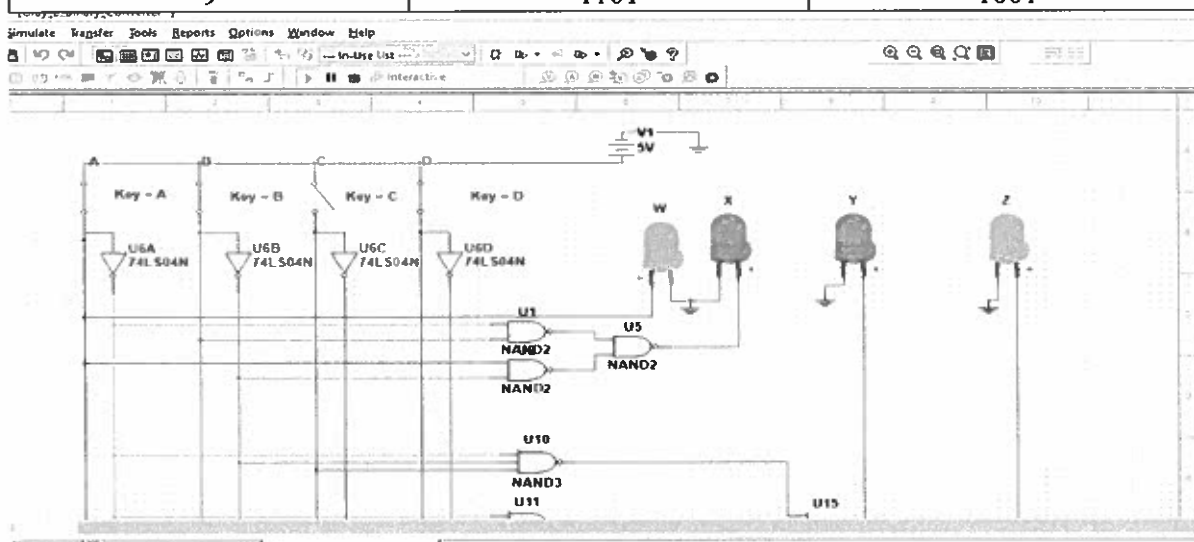


Figure 15: Testing Gray Code to Binary Converter (Gray Code value 1101)

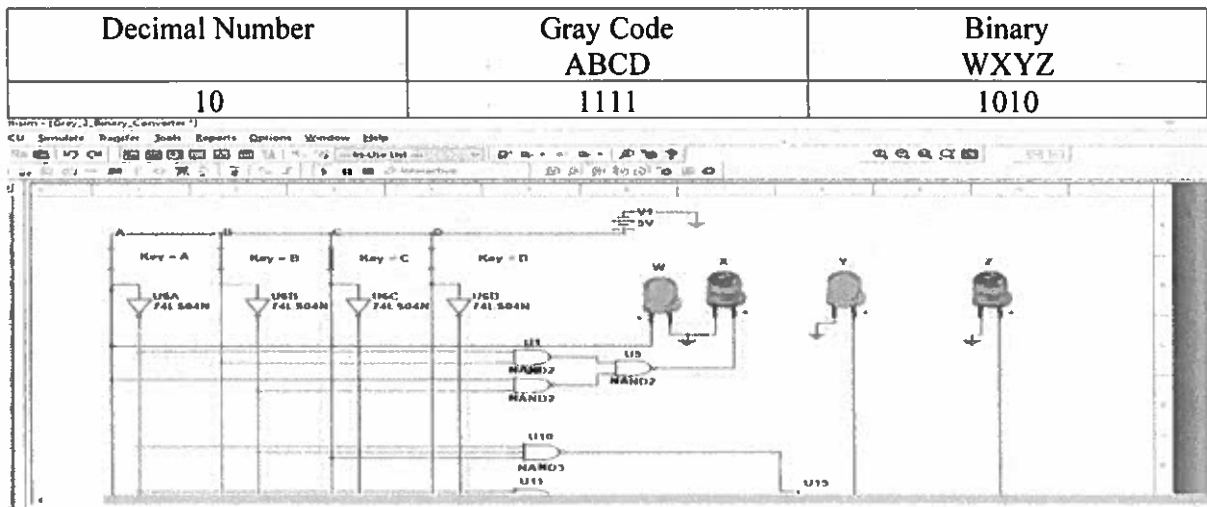


Figure 16: Testing Gray Code to Binary Converter (Gray Code value 1111)

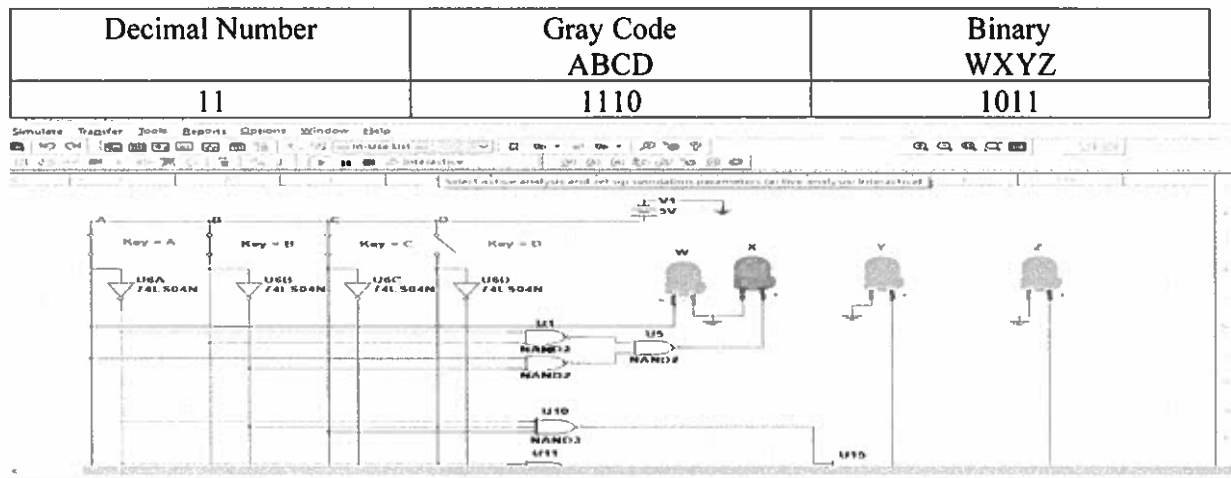


Figure 17: Testing Gray Code to Binary Converter (Gray Code value 1110)

Decimal Number	Gray Code ABCD	Binary WXYZ
14	1001	1110

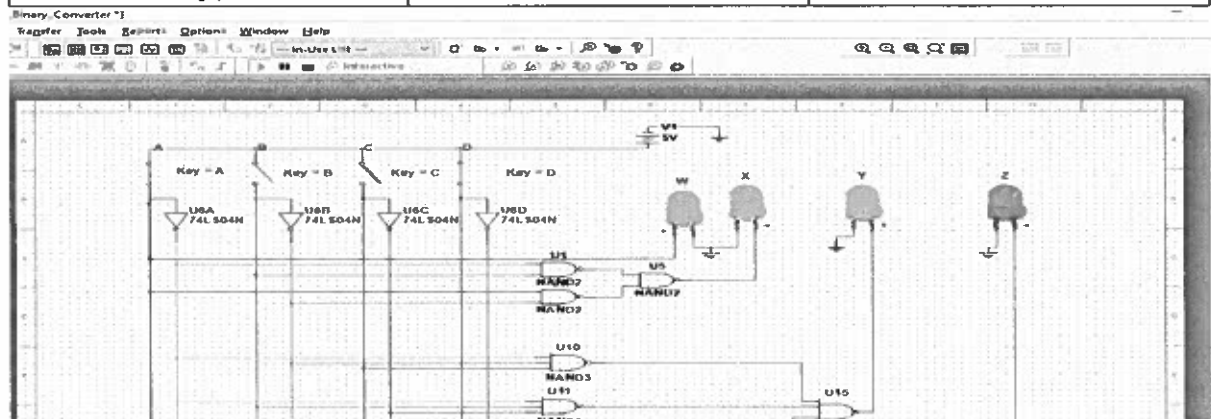


Figure 20.- Testing Gray Code to Binary Converter (Gray Code value 1001)

Decimal Number	Gray Code ABCD	Binary WXYZ
15	1000	1111

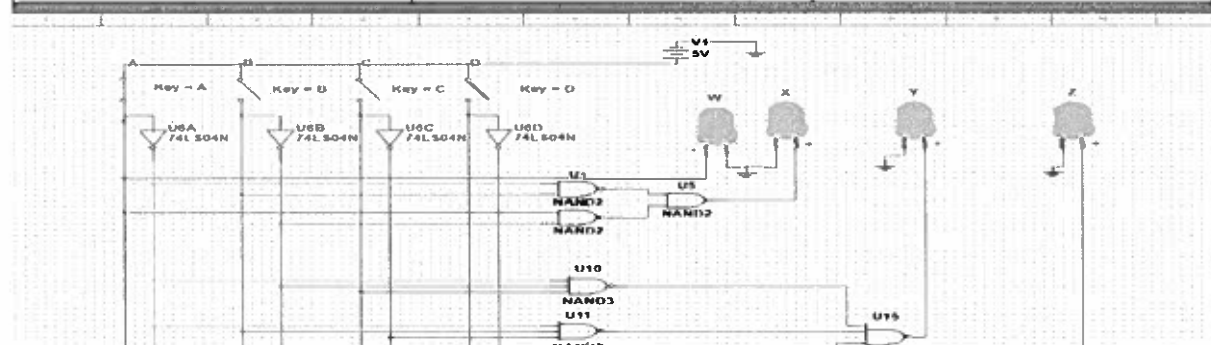


Figure 21: Testing Gray Code to Binary Converter (Gray Code value 1000)

4. Binary to Gray Code Converter Methodology

Design procedures to convert a four-bit Binary number to its equivalent four-bit binary number using exclusive-OR gates required the use of the truth table, K-Maps, Boolean expressions, and Multisim to implement the logic circuit. Each code uses four bits; the converter must have four input variables and four output variables. The four input Binary code variables are represented by the symbols A, B, C, and D, and the four output variables by W, X, Y, and Z correspond to their Gray code values. The truth table relating the input and the output is shown in Table 2. The K-maps (Fig. 1 through 4) are plotted to obtain simplified Boolean functions for the outputs. Each of the four maps represents one of the four outputs of the circuit as a function of the four input variables. From the K-maps, the Boolean expression is obtained. By manipulating the Boolean expression, a logic diagram using exclusive-OR gates was created. The Boolean expressions were then used to implement the logic circuit via Multisim. Testing was conducted by switching the switches (ABCD) to on (1) or off (0) to correspond to its Gray code value from the truth table. The binary values produced by the circuit were then compared to the truth table (table 1). All binary numbers (0-15) converted to Gray code numbers had the correct conversion values.

4.1 Binary to Gray code - Truth Table

Truth Table

Decimal Number	Binary ABCD	Gray Code WXYZ
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Table 2: Truth Table for Binary code to Gray code converter

4.2 Binary to Gray code K-Maps

For W:

$$W = A$$

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

Figure 22: K- Map for Binary to Gray Code converter (Output W)

For X:

$$\begin{aligned} X &= A'B + AB' \\ &= A \oplus B \end{aligned}$$

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

Figure 23: K- Map for Gray Code to Binary converter (Output X)

For Y:

$$\begin{aligned} Y &= BC' + B'C \\ &= B \oplus C \end{aligned}$$

AB \ CD	00	01	11	10
00			1	1
01	1	1		
11	1	1		
10			1	1

Figure 24: K- Map for Binary to Gray Code converter (Output Y)

For Z:

$$Z = C'D + CD'$$

$$= C \oplus D$$

AB \ CD	00	01	11	10
00				
01				
11				
10				

Figure 25: K- Map for Binary to Gray Code converter (Output Z)

Output Boolean expression:

$$W = A$$

$$X = A'B + AB' = A \oplus B$$

$$Y = BC' + B'C = B \oplus C$$

$$Z = C'D + CD' = C \oplus D$$

4.3 Binary to Gray Code converter - Logic Circuit Implementation

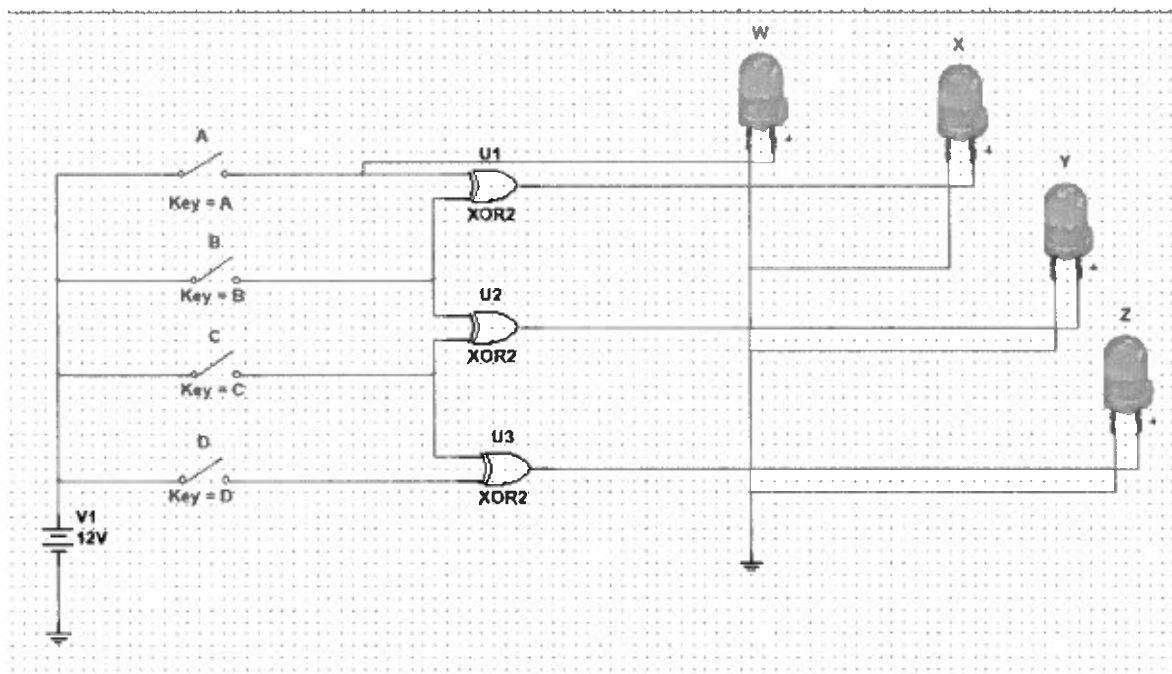


Figure 26: Binary to Gray Code converter - Logic Circuit Implementation

4.4 Testing Binary to Gray code converter

Testing was conducted by switching the switches (ABCD) to on (1) or off (0) to correspond to its binary values from the truth table. The Gray code values (WXYZ) was linked to a LED. LED that was on corresponded to a value of 1, while if the LED remained off the value corresponded to the value 0. The Gray code values (WXYZ) produced by the logic circuit was then compared to the truth table (table 2). All Binary values were checked and produced the correct values.

Testing Binary to Gray code Converter:

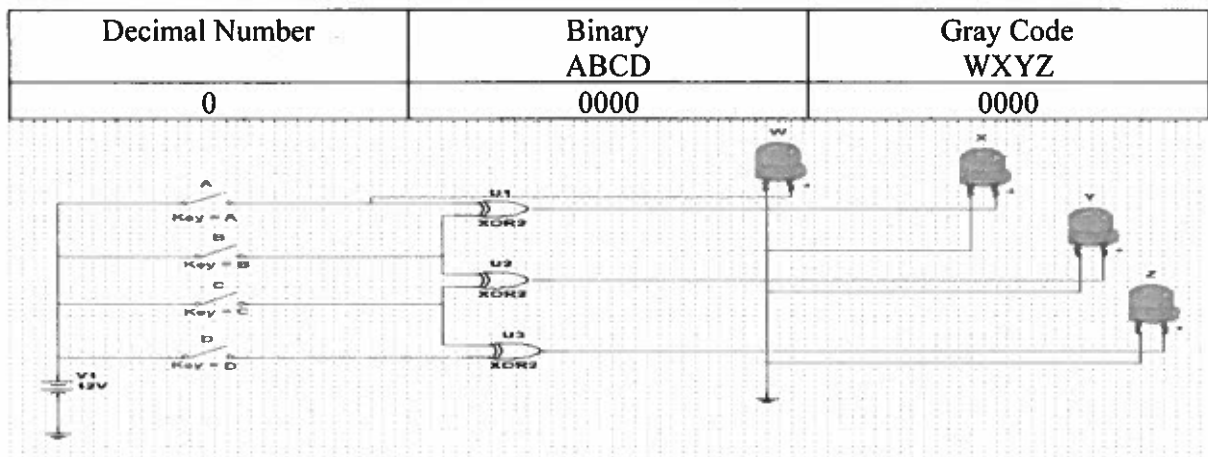


Figure 27: Testing Binary to Gray Code Converter (Binary value 0000)

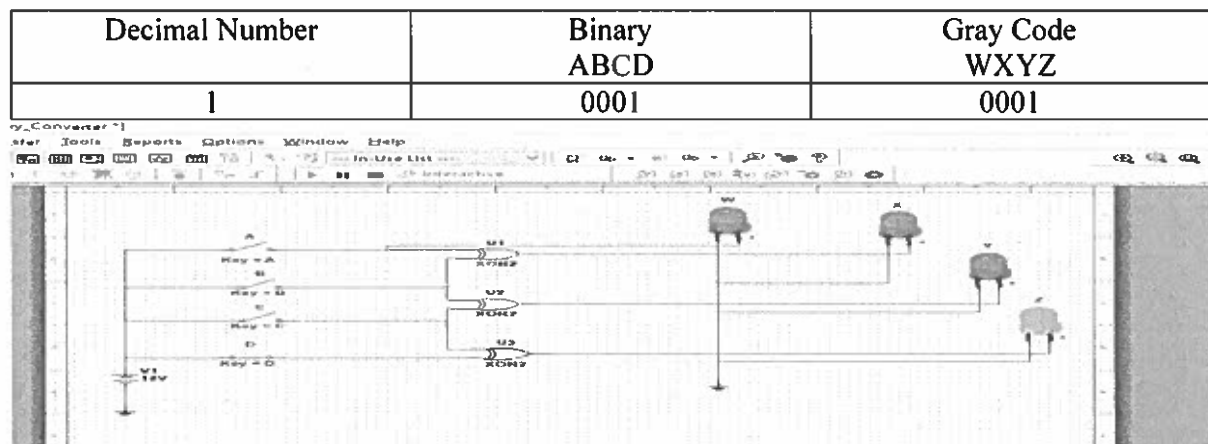


Figure 28: Testing Binary to Gray Code Converter (Binary value 0001)

Decimal Number	Binary ABCD	Gray Code WXYZ
2	0010	0011

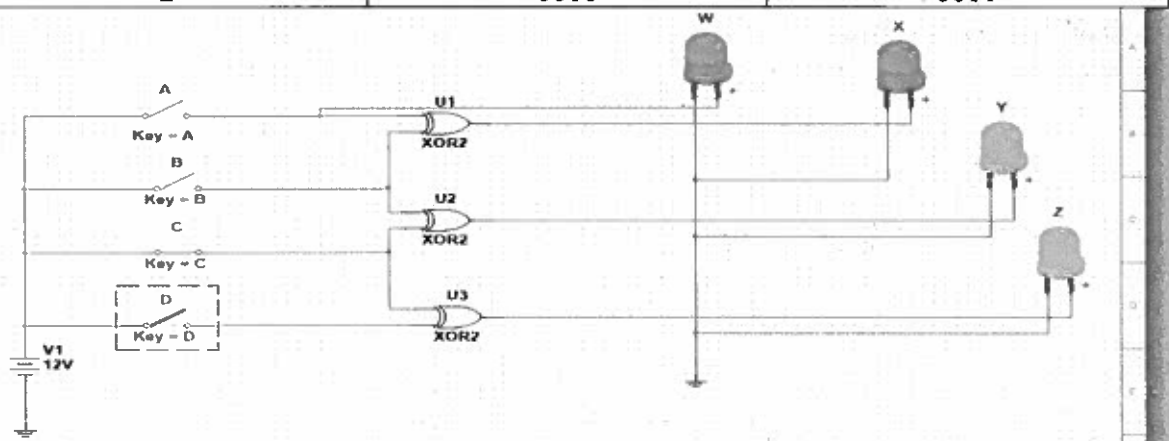


Figure 29: Testing Binary to Gray Code Converter (Binary value 0010)

Decimal Number	Binary ABCD	Gray Code WXYZ
3	0011	0010

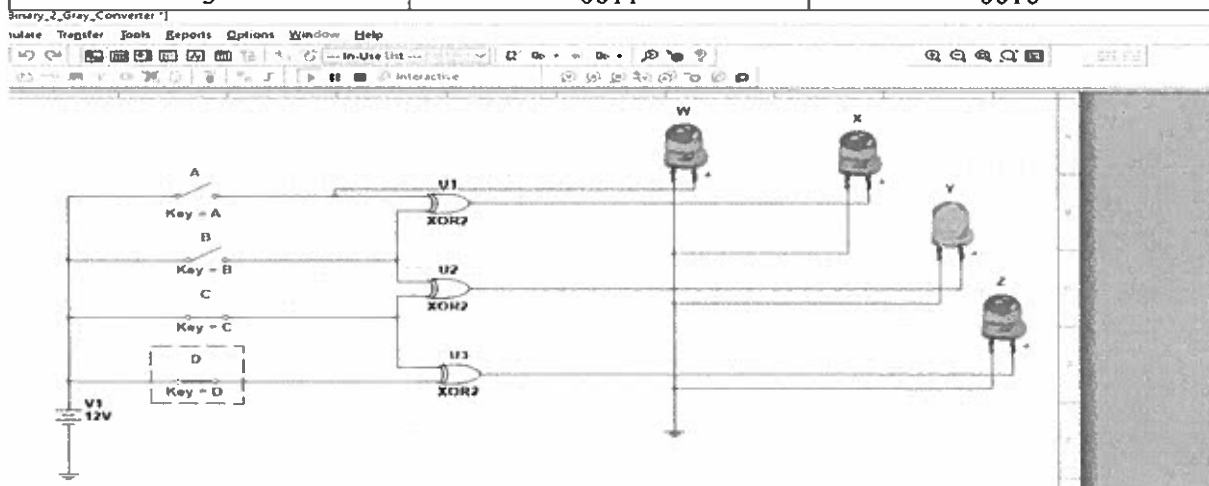


Figure 30: Testing Binary to Gray Code Converter (Binary value 0011)

Decimal Number	Binary ABCD	Gray Code WXYZ
4	0100	0110

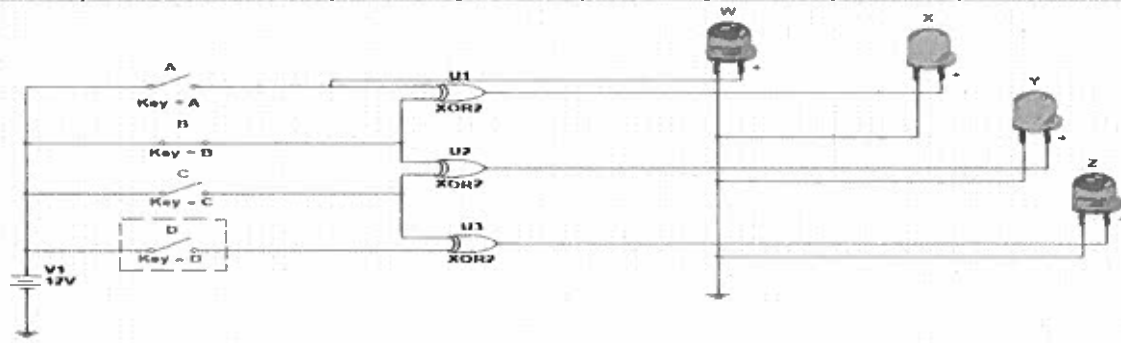


Figure 31: Testing Binary to Gray Code Converter (Binary value 0100)

Decimal Number	Binary ABCD	Gray Code WXYZ
5	0101	0111

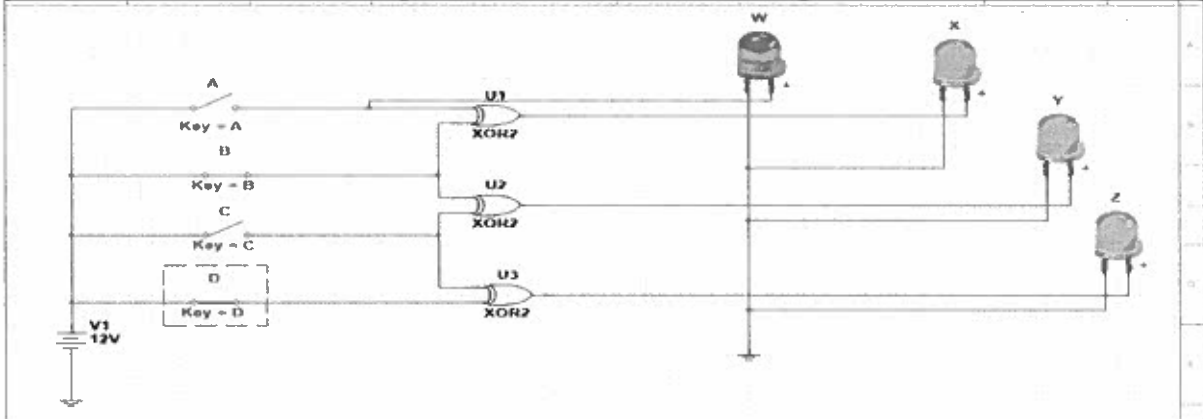


Figure 32: Testing Binary to Gray Code Converter (Binary value 0101)

Decimal Number	Binary ABCD	Gray Code WXYZ
6	0110	0101

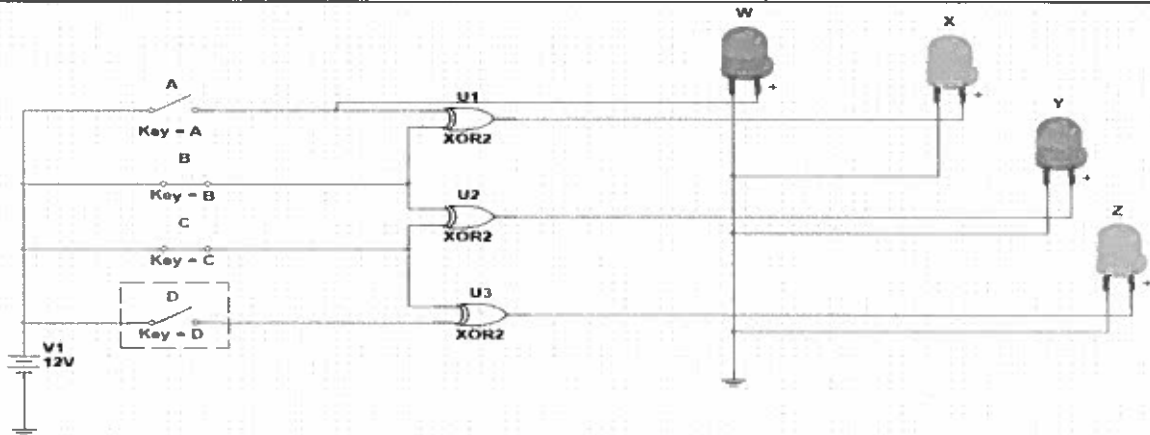


Figure 33: Testing Binary to Gray Code Converter (Binary value 0110)

Decimal Number	Binary ABCD	Gray Code WXYZ
7	0111	0100

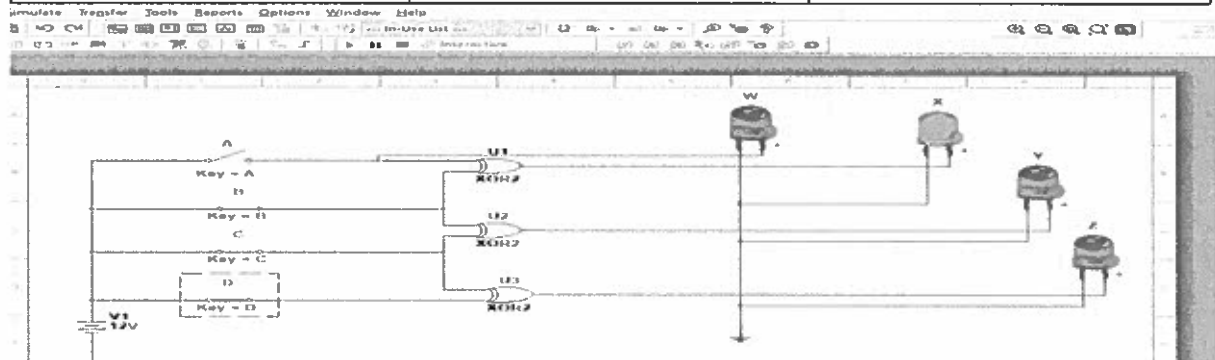


Figure 34: Testing Binary to Gray Code Converter (Binary value 0111)

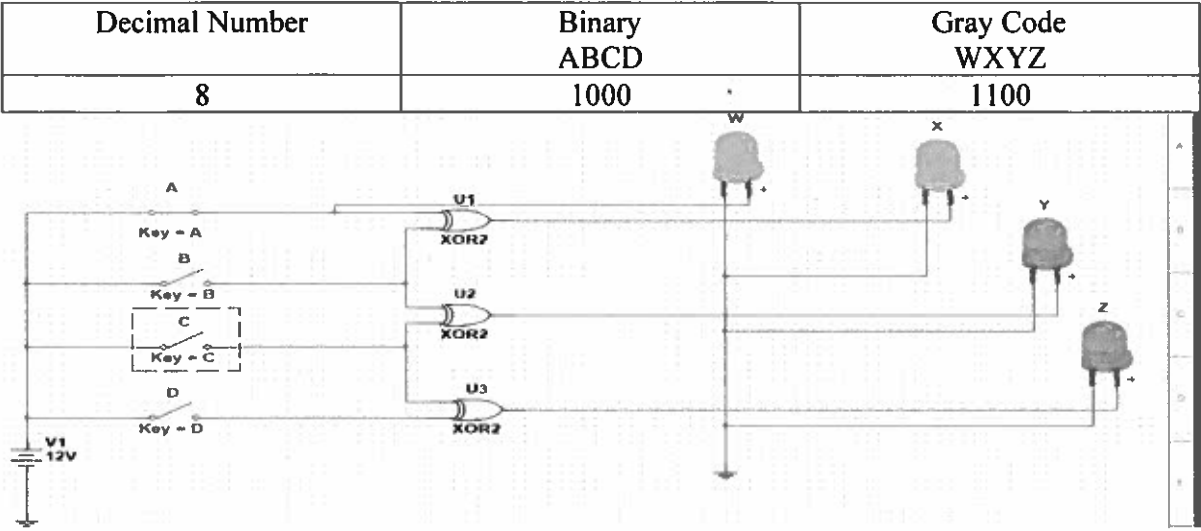


Figure 35: Testing Binary to Gray Code Converter (Binary value 1000)

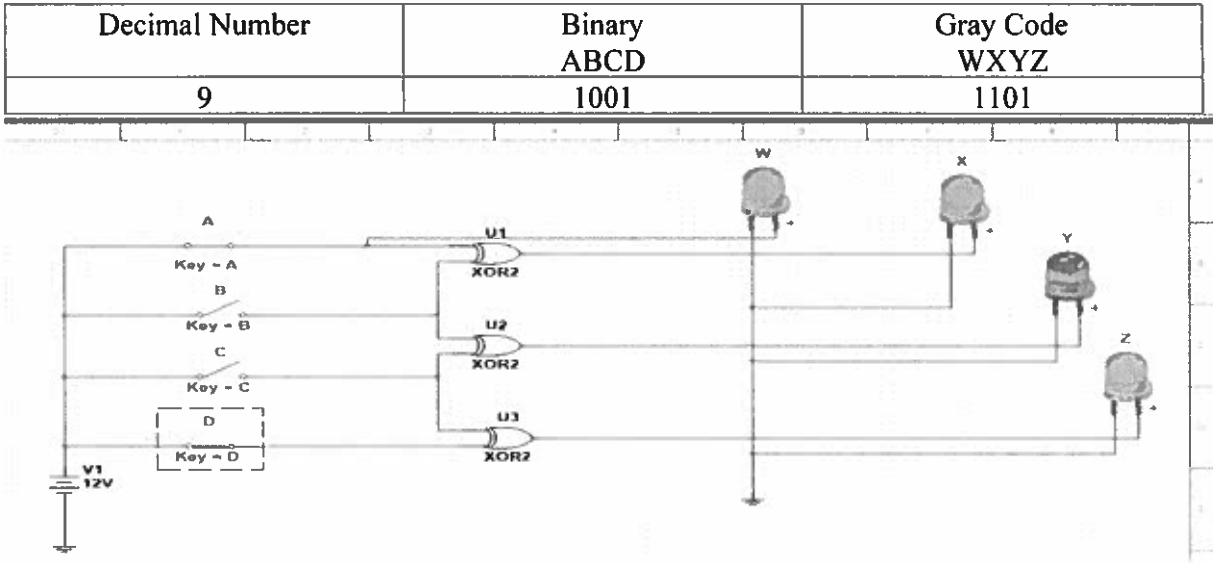


Figure 36: Testing Binary to Gray Code Converter (Binary value 1001)

Decimal Number	Binary ABCD	Gray Code WXYZ
10	1010	1111

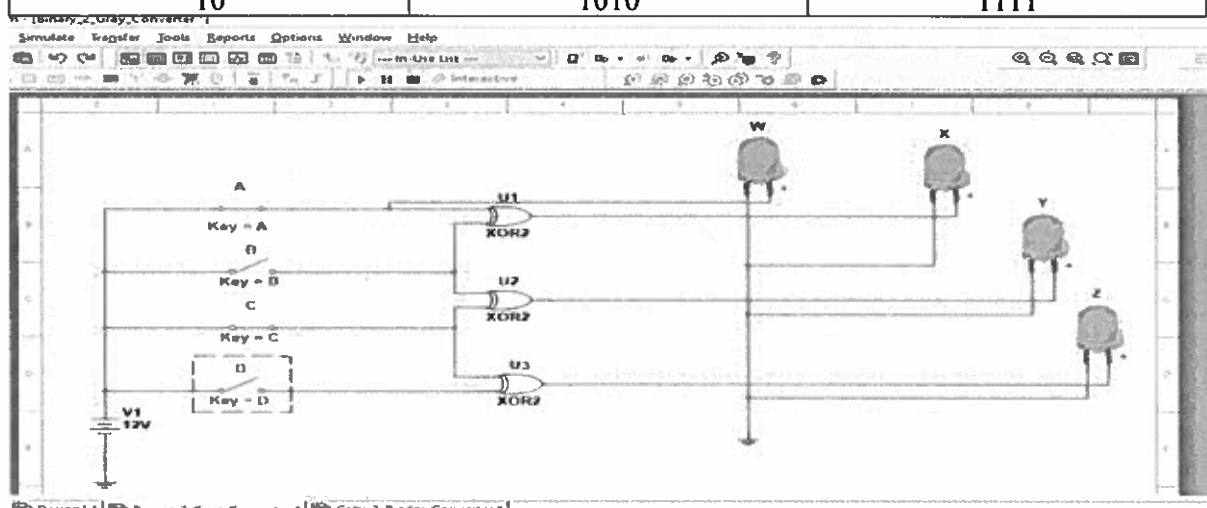


Figure 37: Testing Binary to Gray Code Converter (Binary value **1010**)

Decimal Number	Binary ABCD	Gray Code WXYZ
11	1011	1110

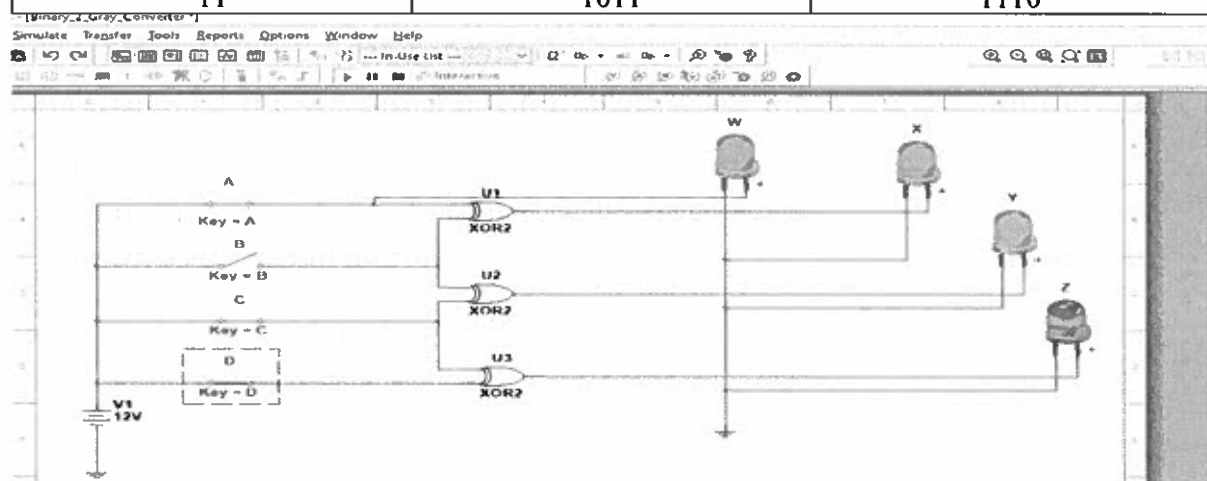


Figure 38: Testing Binary to Gray Code Converter (Binary value **1011**)

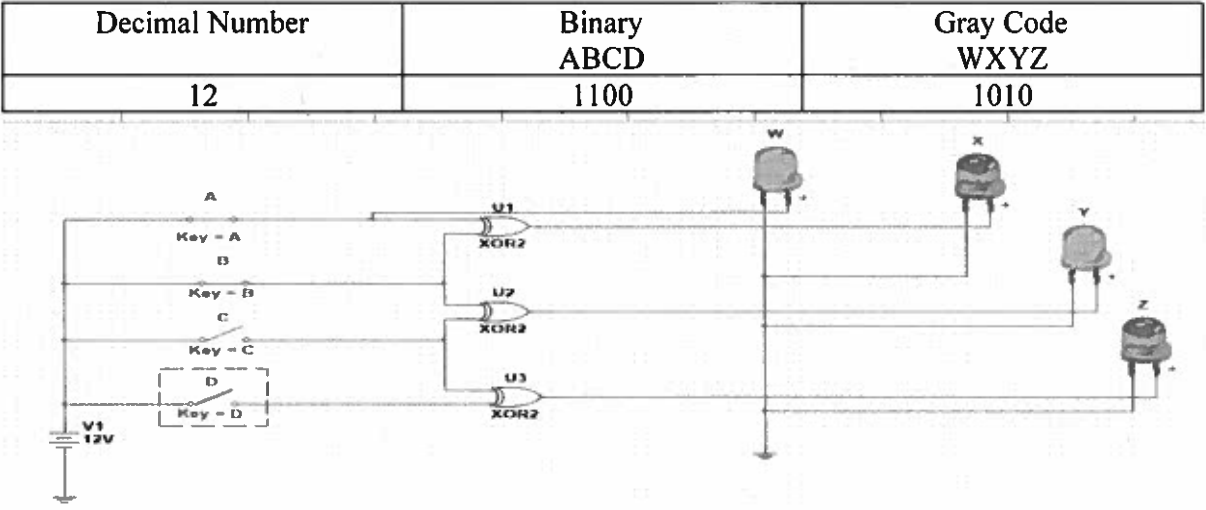


Figure 39: Testing Binary to Gray Code Converter (Binary value 1100)

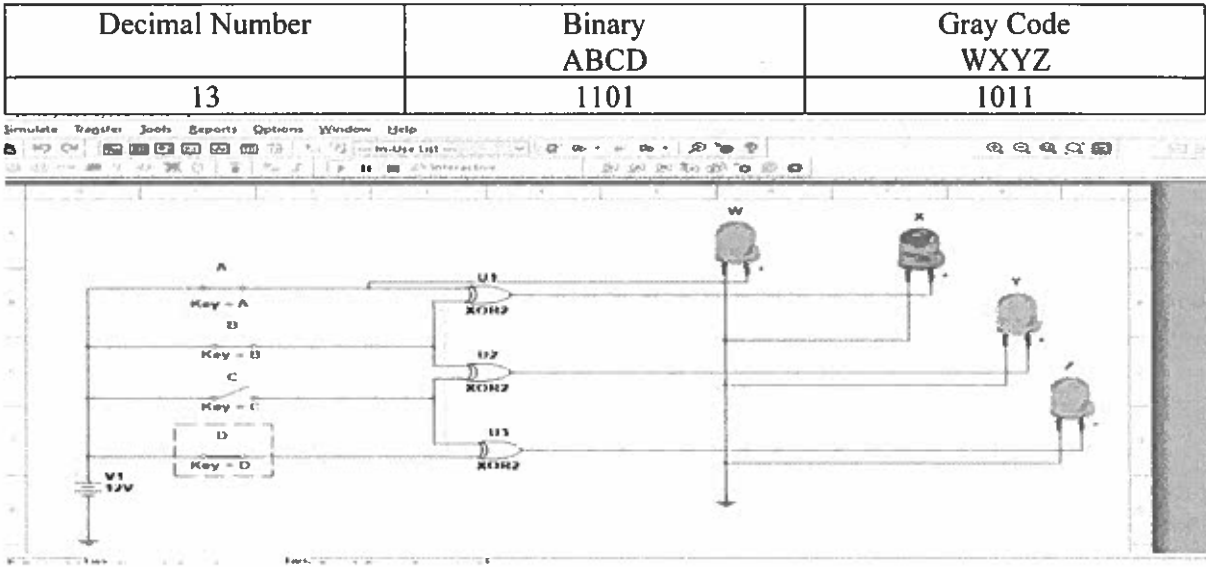


Figure 40: Testing Binary to Gray Code Converter (Binary value 1101)

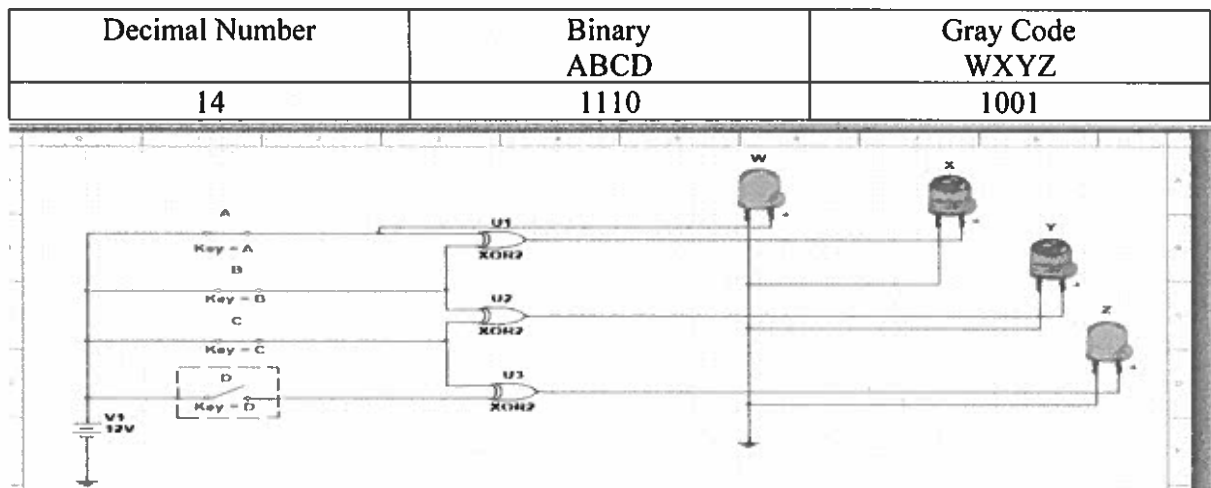


Figure 41: Testing Binary to Gray Code Converter (Binary value 1110)

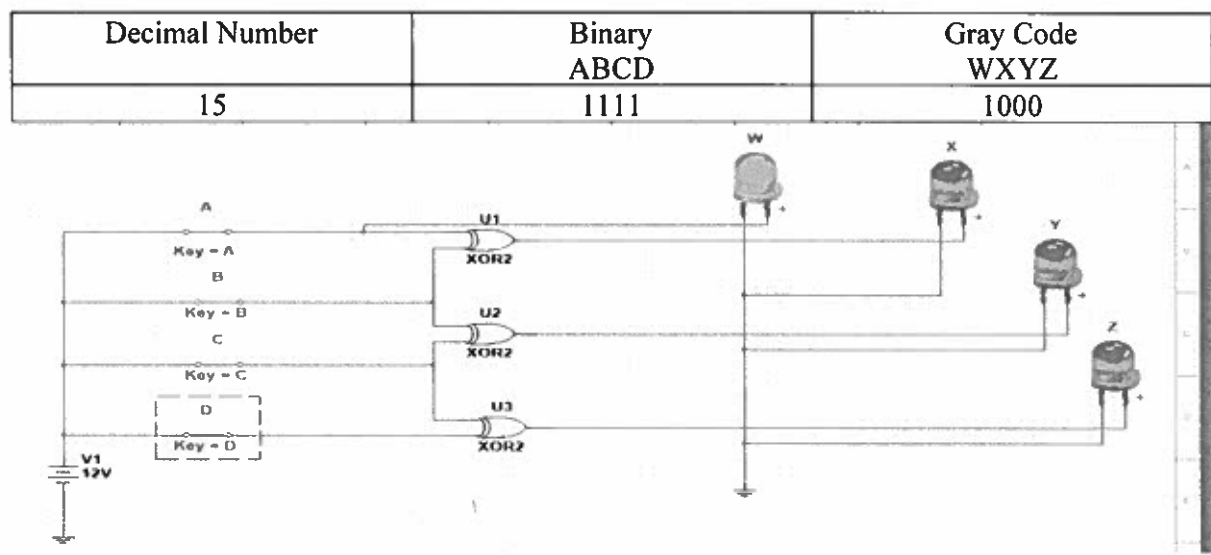


Figure 42: Testing Binary to Gray Code Converter (Binary value 1111)

5. Results

It may be more beneficial to use codes like binary, octal, hexadecimal, Binary Coded Decimal (BCD), Excess-3, or Gray code in different situations. Numbers where all the bits flip, for example, 011 to 100, may produce an incorrect number. Gray code addresses this problem since only one bit changes its value during the transition. A universal gate is a gate that can implement any Boolean function without the necessity of any other gate type. The NAND and ~~exclusive-OR~~ gates are examples of universal gates. Both the NAND and exclusive-OR gates have tremendous advantages for their economical and ease of fabrication. In the design process, progression from truth table, k-maps, Boolean expression, and logic circuit implementation via Multisim highlighted steps taken for code conversion.

6. Conclusion

In this project, two code converters were designed, implemented, and tested. Using NAND gates, the first code converter took a four-bit Gray code number and converted it to its equivalent four-bit binary number. The second code converter that was designed converts a four-bit binary number into the equivalent four-bit Gray code using exclusive-OR gates. Both NAND and exclusive-OR gates are universal gates. The power and usefulness of universal gates help in implantation due to their economic and ease of use. The project explored the importance of non-weighted binary code, combinational circuit design procedures, and universal gates' significance.

References

- [1] M. M. Mano and M. D. Ciletti, *Digital design with an introduction to the Verilog HDL, VHDL, and systemverilog*. New York, NY: Pearson, 2019.

AVERAGE

88



Digital Design Project 1

Tri'Heem Neville

Introduction

In this project, I will design a four-bit grey code number to a four-bit binary number convertor and a four-bit binary number to a four-bit grey code number convertor. When making the grey code to binary convertor, I had to implement the circuit using the universal NAND gate. When making the binary to grey code convertor, I had to implement the circuit using the XOR gates. To make the convertor I had to use my knowledge of the design procedure, and a Multisim simulation of the convertor was required. I will walk through the design process and how it translates into the Multisim simulation.

Design Process

The design process for combinational circuits starts with determining the number of inputs and outputs. In the case of making the convertor circuits, there are four inputs and four outputs because the circuit is converting a four-bit grey code number to a four-bit binary number and vice versa, as outlined in the project requirements. The second step of the design process is deriving the truth table that shows the relationship between inputs and outputs. The third step of the design process is obtaining the simplified Boolean expression for each output using the truth table and K-maps. The fourth and final step of the design process is drawing the logic diagram to make sure the design is correct. I did this using a Multisim simulation which will be outline in the next section to show the correctness of the convertor circuits. The figures below represent the design process for the four-bit grey code number to four-bit binary number convertor.

	Input				Output				T
	A	B	C	D	W	X	Y	Z	
0	0	0	0	0	0	0	0	0	
1	0	0	0	1	0	0	0	1	
2	0	0	1	1	0	0	1	0	
3	0	0	1	0	0	0	1	1	
4	0	1	1	0	0	1	0	0	
5	0	1	1	1	0	1	0	1	
6	0	1	0	1	0	1	1	0	
7	0	1	0	0	0	1	1	1	
8	1	1	0	0	1	0	0	0	
9	1	1	0	1	1	0	0	1	
10	1	1	1	1	1	0	1	0	
11	1	1	1	0	1	0	1	1	
12	1	0	1	0	1	1	0	0	
13	1	0	1	1	1	1	0	1	
14	1	0	0	1	1	1	1	0	
15	1	0	0	0	1	1	1	1	

Figure 1.1 Grey code to binary truth table

		W				
AB	CD	00	01	11	10	
		0	0	0	0	$W = A$
0	1	0	0	0	0	
1	1	1	1	1	1	A'
1	0	1	1	1	1	

Figure 1.2 Simplified Boolean expression for output W

X

AB \ CD	00	01	11	10	
00	0	0	0	0	
01	1	1	1	1	$A'B$
11	0	0	0	0	
10	1	1	1	1	AD'

$X = A'B + AB'$

Figure 1.3 Simplified Boolean expression for output X

Y

AB \ CD	00	01	11	10	
00	0	1	1	0	$A'B'C$
01	1	1	0	0	$A'BC'$
11	0	0	1	1	ABC
10	0	0	0	1	$AB'C'$

$Y = A'B'C + A'BC' + ABC + AB'C'$

Figure 1.4 Simplified Boolean expression for output Y

Z

AB \ CD	00	01	11	10	
00	0	1	0	1	$A'B'C'D'$
01	1	0	0	0	$A'BCD$
11	0	1	0	1	$ABC'D'$
10	1	0	1	0	$AB'CD$

$Z = A'B'C'D' + A'BCD + ABC'D' + AB'CD$

Figure 1.5 Simplified Boolean expression for output Z

The figures below represent the design process for the four-bit binary number to four-bit grey code number convertor.

	Input				Output			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Figure 2.1 Binary to grey code truth table

AB \ CD		W			
		00	01	11	10
00		0	1	3	2
01		4	5	7	6
11		11	13	15	14
10		8	9	11	10

A

$W = A$

Figure 2.2 Simplified Boolean expression for output W

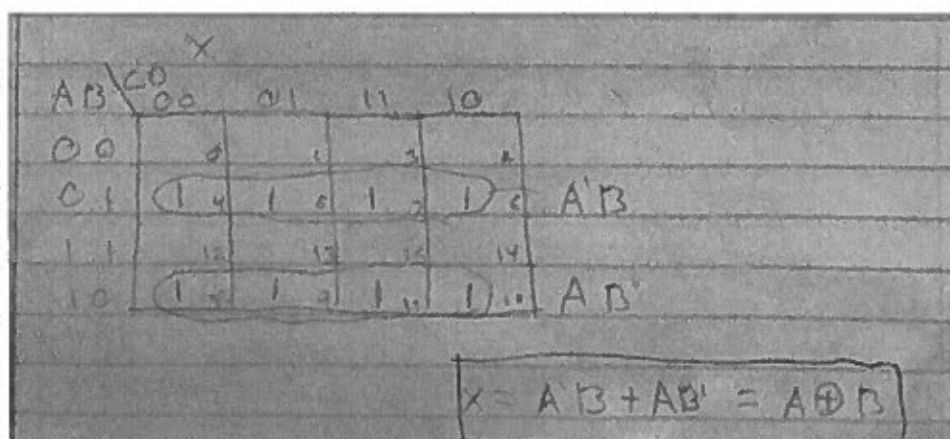


Figure 2.3 Simplified Boolean expression for output X

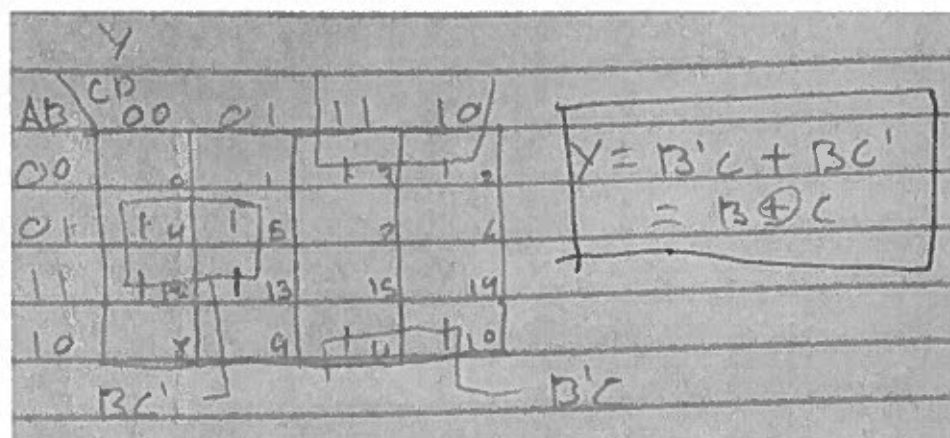


Figure 2.4 Simplified Boolean expression for output Y

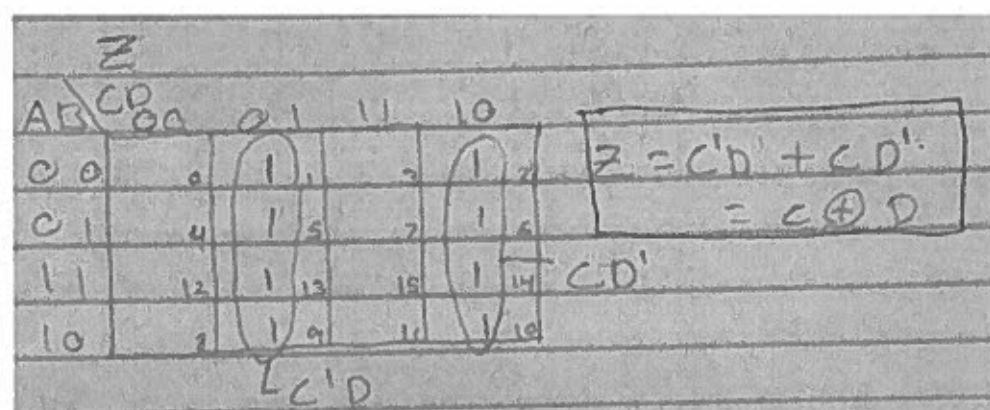


Figure 2.5 Simplified Boolean expression for output Z

Multisim Simulation

After completing the design process, I was then able to build the circuits on Multisim to test their functionality which is the last step in the design process. To build the four-bit grey code to four-bit binary number convertor on Multisim, I needed 12 NAND gates, four inputs, and four outputs. In this section I will show that the simulation is functioning properly and agrees with the truth table. The conversions that I test for both the grey code convertor and binary code convertor are highlighted in the truth tables in figures 1.1 and figure 2.1. Below is the grey code to binary convertor without any inputs selected.

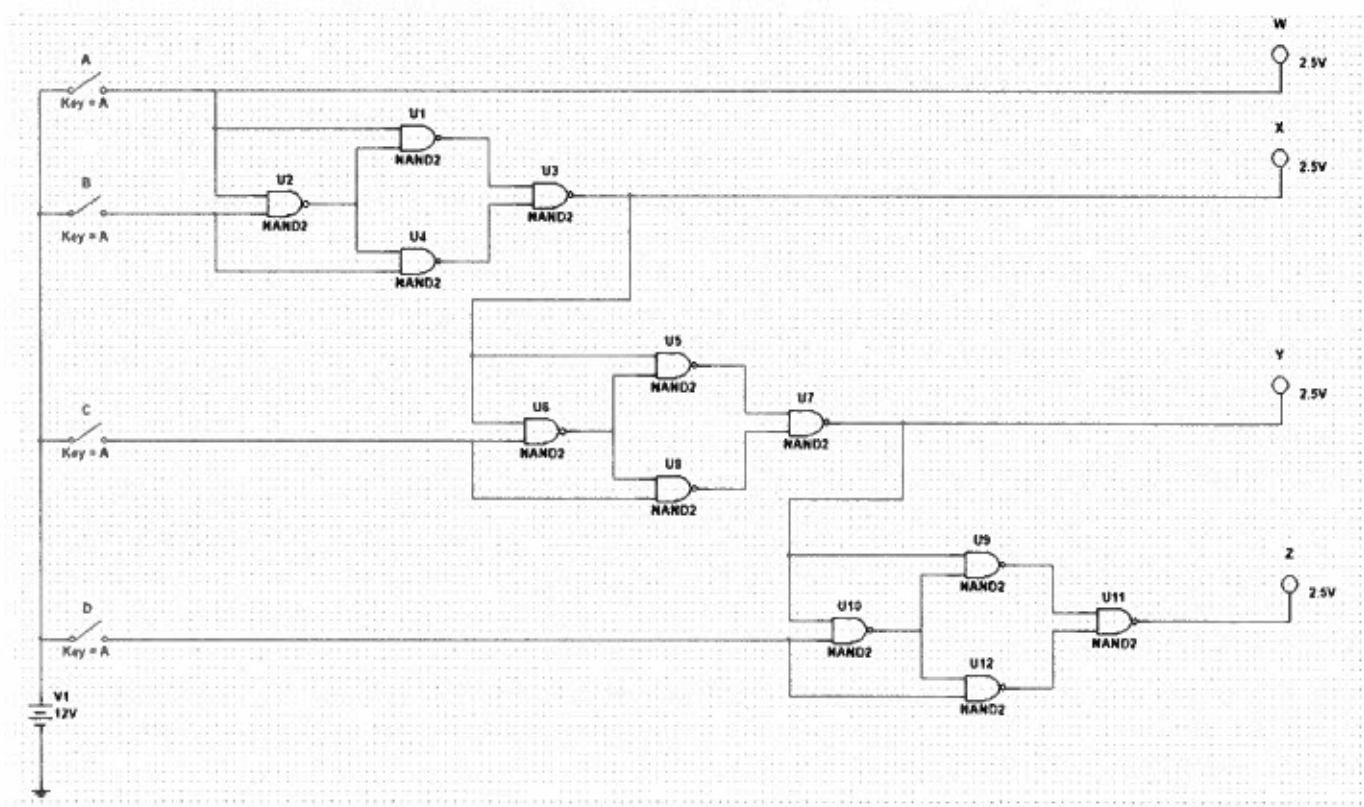


Figure 3.1 Grey code to Binary Convertor

The figures below are showing an example of converting grey code input to binary output.

When the switch on the left is open the input is 0 and when it is closed the input is 1. For the output, when the light is green the output is 1 and when the light is not on the output is 0.

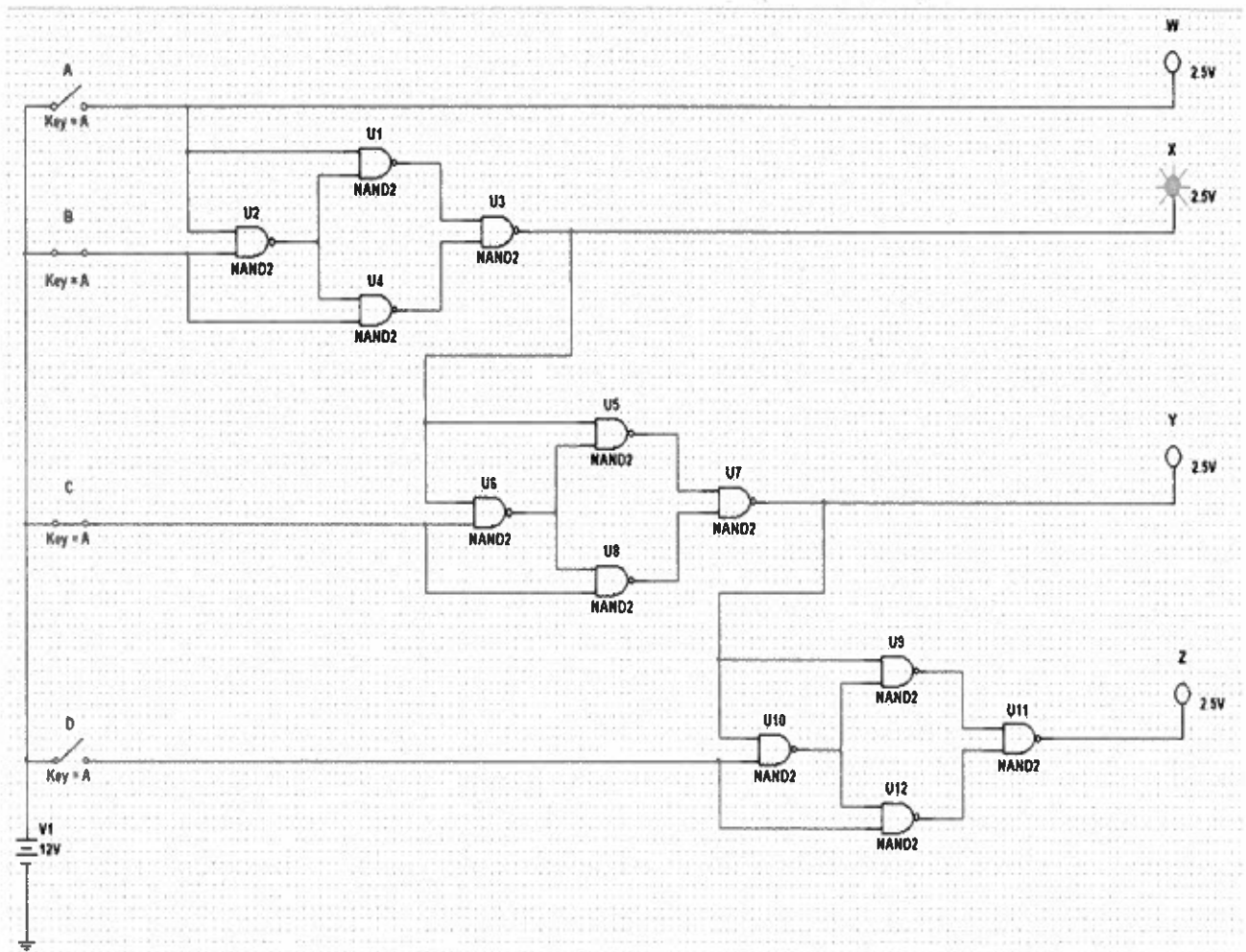


Figure 3.2 Grey code input is 0110 and binary output is 0100 as shown in figure 1.1

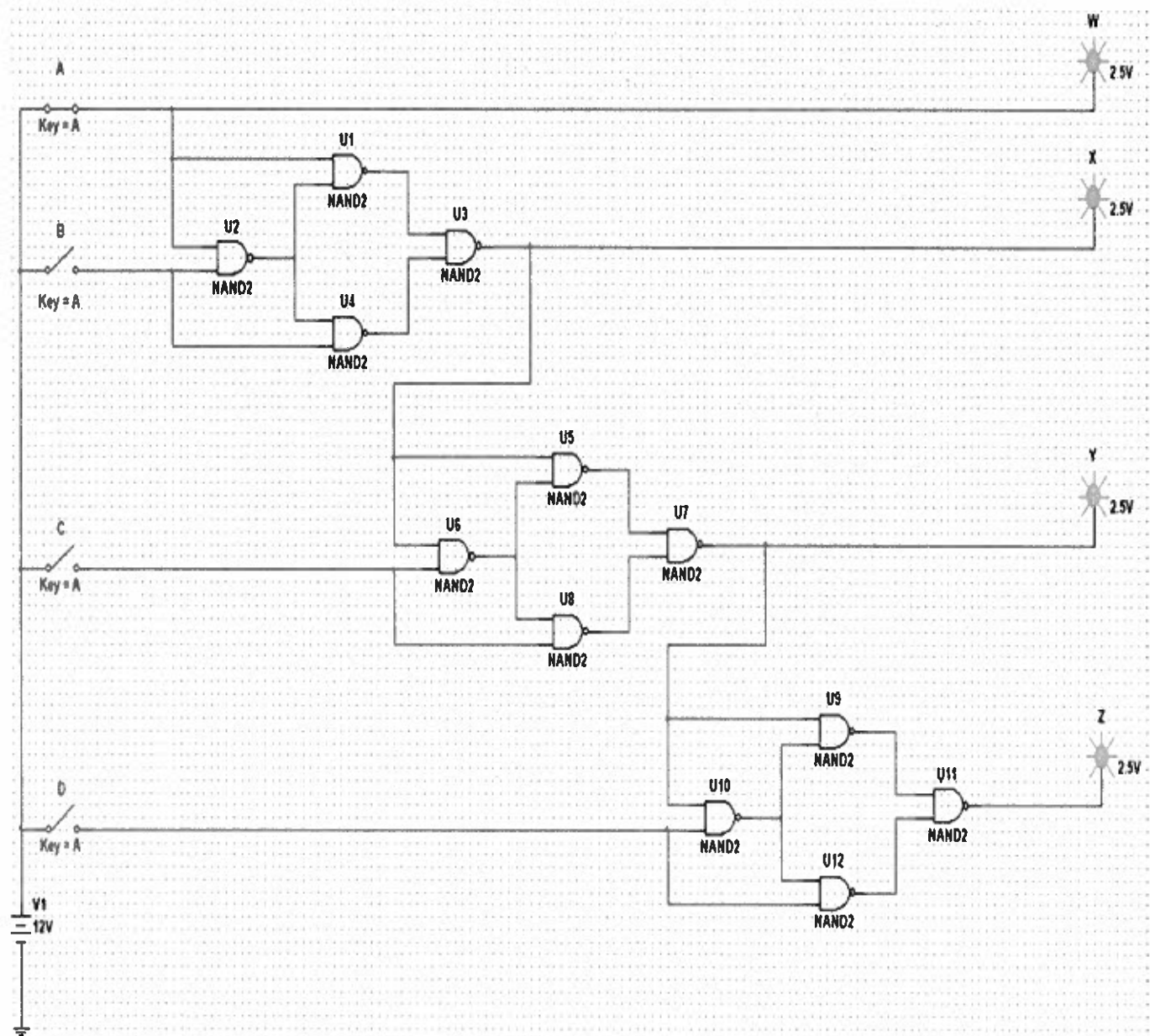


Figure 3.3 Grey code input 1000 and binary output 1111 as shown in figure 1.1

To build the binary number to grey code convertor on Multisim, I needed three XOR gates, four inputs, and four outputs. The figure below is the binary to grey code convertor with no inputs selected.

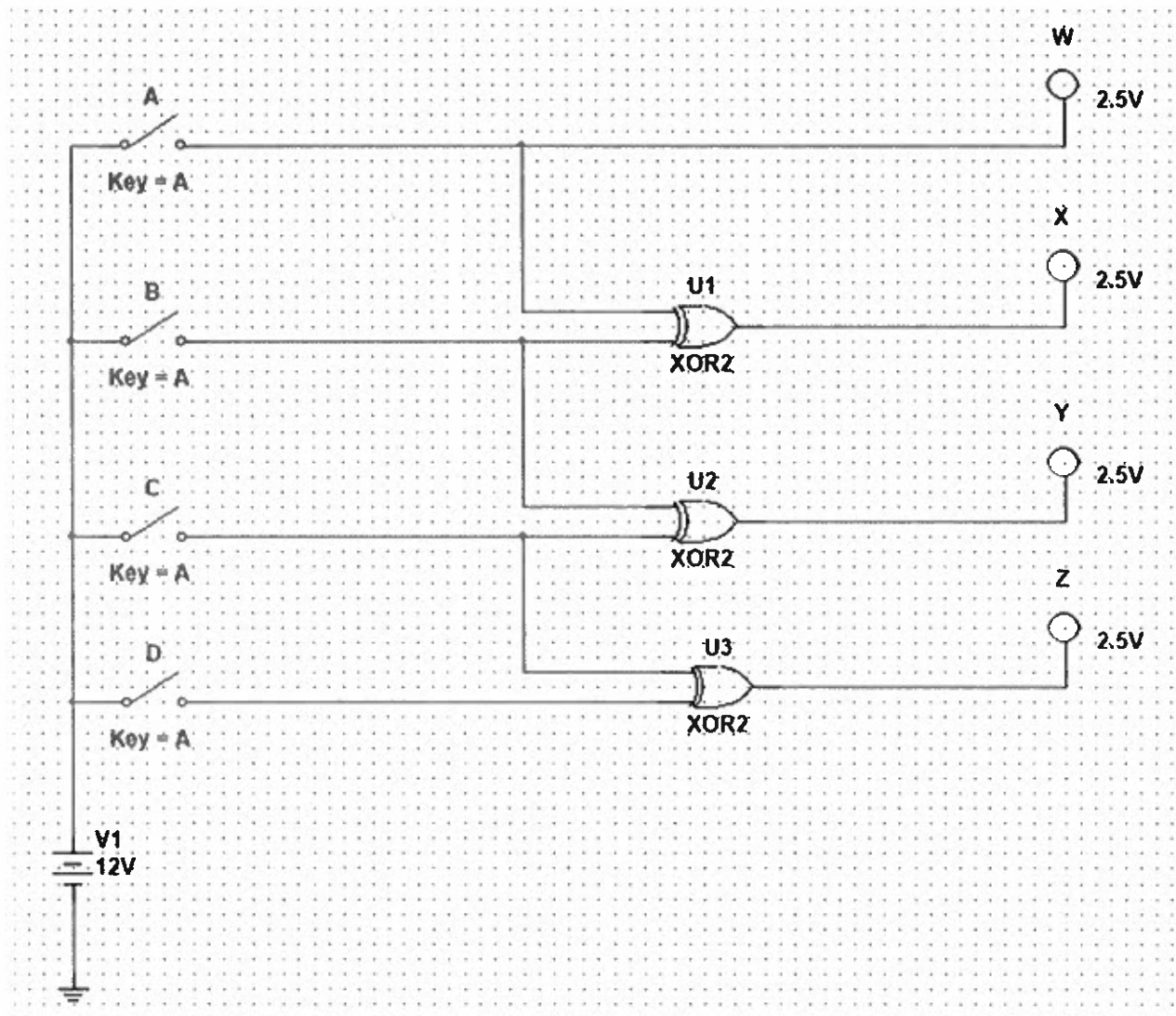


Figure 4.1 Binary to Grey Code Convertor

The figures below are showing an example of converting binary input to grey code output.

When the switch on the left is open the input is 0 and when it is closed the input is 1. For the output, when the light is green the output is 1 and when the light is not on the output is 0.

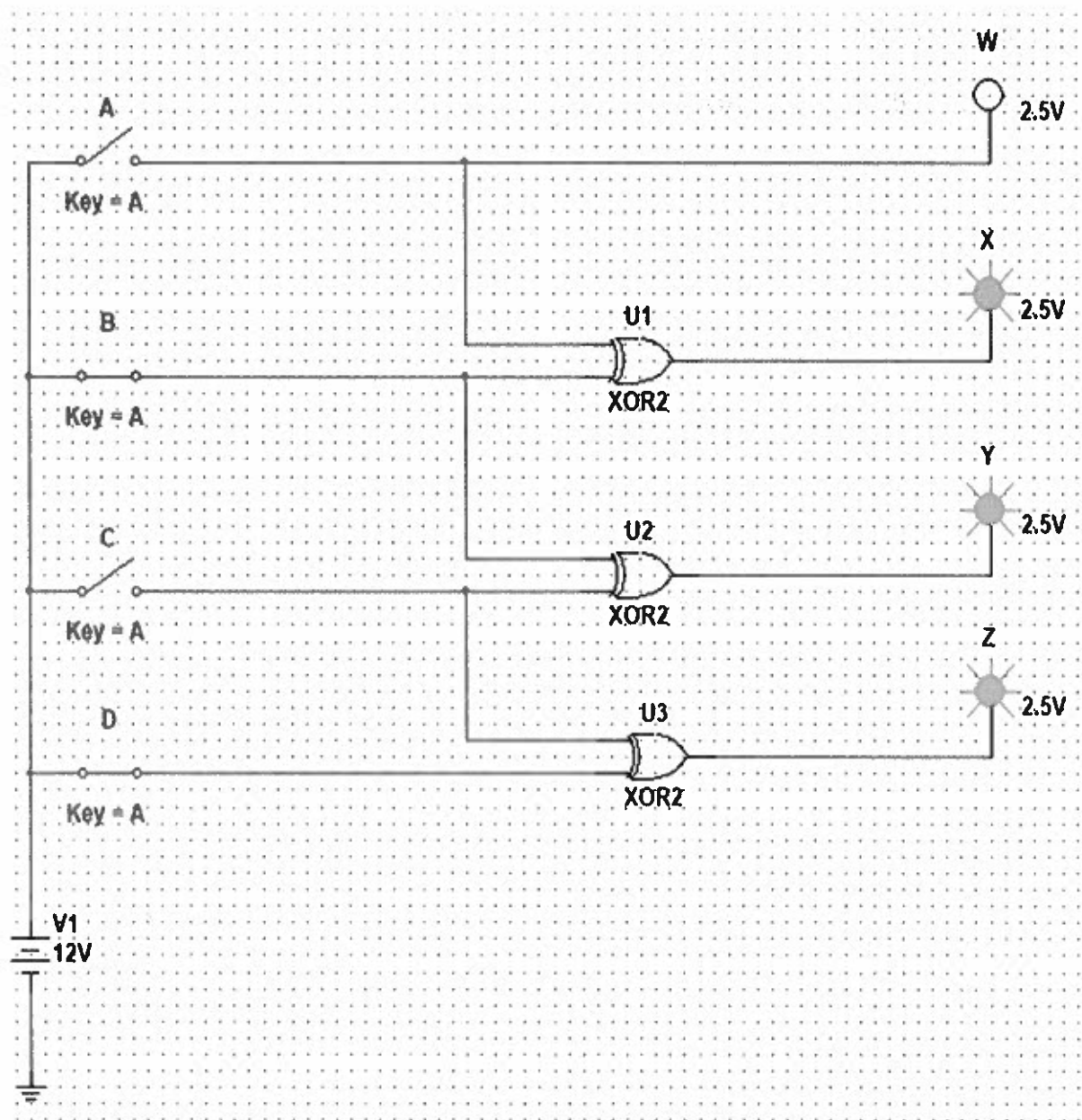


Figure 4.2 Binary input 0101 and grey code output 0111 as shown in figure 2.1

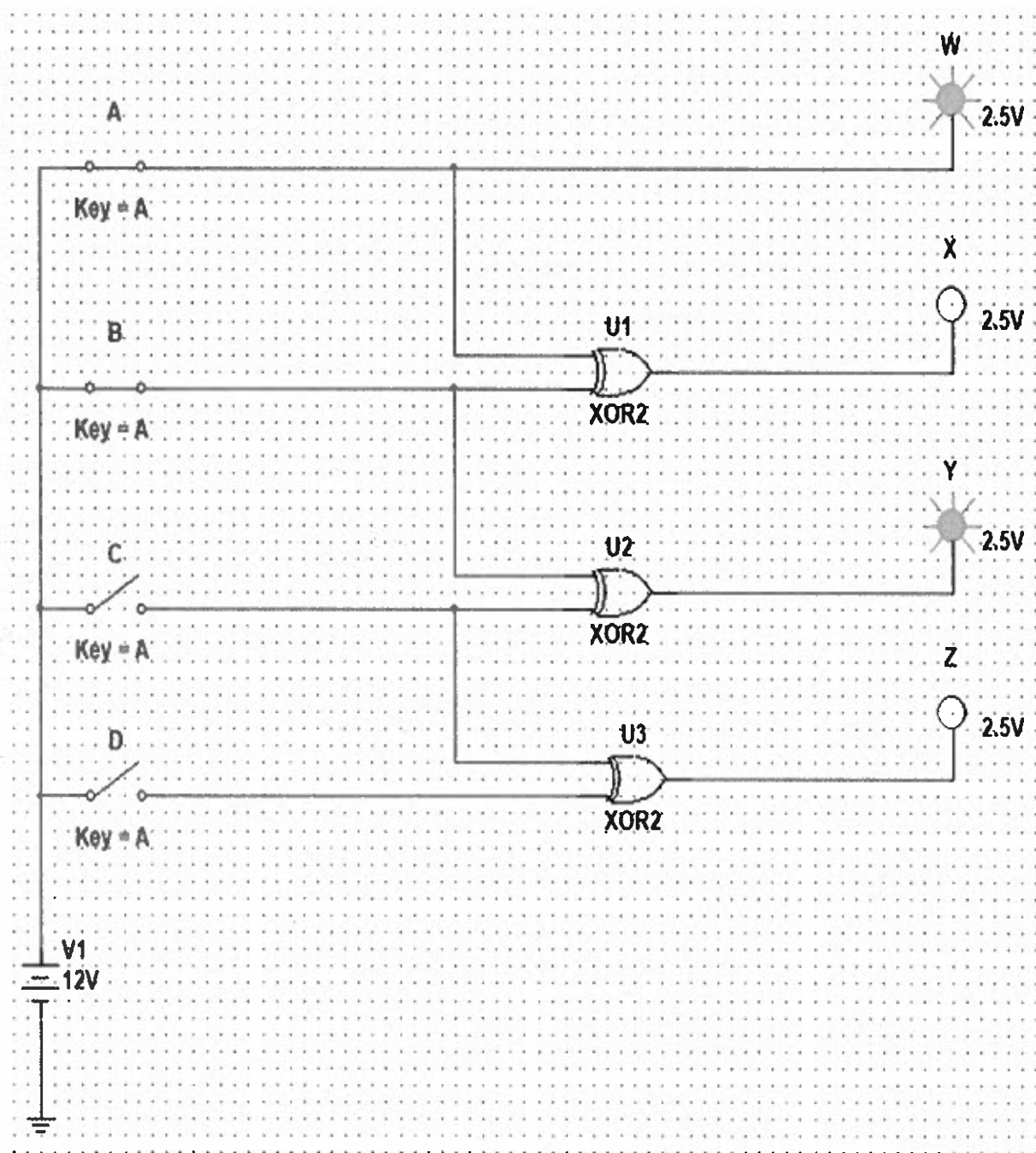
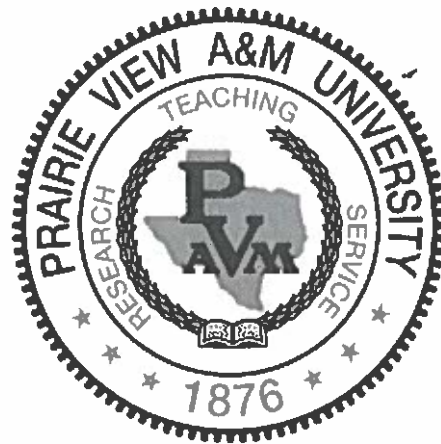


Figure 4.3 Binary input 1100 and Grey Code output 1010 as shown in figure 2.1

Conclusions

In conclusion the project was completed successfully, and the convertor circuits are functioning as intended as shown in the Multisim simulation. Code convertors are necessary because they can make two systems compatible even though they use different codes. The grey code is a way to order the binary system of zeros and ones such that two values differ in only one bit, a bit is also referred to as a binary digit. There are two universal gates NAND and NOR, they are important because they can implement all of the logic functions OR, AND, and NOT. The grey code to binary number convertor uses one of the universal gates in combinational logic circuits, the NAND gate.

BELOW AVERAGE



78

Introduction to Digital Design

Course: ELEG 4330

Dr. Xiangfang Li

Combinational Logics Design: Code Converters
Project #1

Submitted By:
Submission Date:

Javian Walker
11/8/2021

**Department of Electrical and Computer Engineering
Prairie View A&M University**

Project Aim:

To realize Binary to Gray code converter and vice versa.

Learning Objective:

1. To learn the importance of non-weighted binary code.
2. To get familiar with combinational circuits design procedure.
3. Understand the importance of Universal gate.

Project requirements:

Design a combinational circuit with four inputs and four outputs:

(1) That convert a four-bit Gray code number (textbook table 1.6) into the equivalent four-bit binary number. Please implement the circuit with Universal gate (NAND gates).

(2) That convert a four-bit binary number into the equivalent four-bit Gray code. Please implement the circuit with exclusive-OR gate.

Procedure:

(1) Please demonstrate the design process, from truth table, K-map, corresponding Boolean expressions, and finally logic circuits implementation.

(2) Please discuss the following questions in your report. (a)What is the necessity of code conversions? (b)What is gray code? (c)What are universal gates? Why it is important?

(3) If you could access the lab, you could physically implement the logic using appropriate ICs. If you could not access the lab, you could use Multisim to simulate the logic circuits.

(4) Please test whether your design is correct, and demonstrate the testing procedure in your report.

In your report, please based on the knowledge you learned in the class, demonstrate your design process to produce the needed result with the consideration of how to make the design more optimal, more efficient, and at the same time with economical consideration and limitation.

• Please demonstrate the design process, from truth table, K-map, corresponding Boolean expressions, and finally logic circuits implementation.

-Truth Table:

• 4 bit Binary to Gray Code (Truth Table):

B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

-K-map:

• 4 bit Binary to Gray Code (K-maps)

		A_3						A_2			
$B_3 B_2$	$B_3 B_2$	00	01	11	10	$B_3 B_2$	$B_3 B_2$	00	01	11	10
		0	1	3	2			0	1	3	2
00	00	0	1	3	2	00	00	0	1	3	2
01	01	4	5	7	6	01	01	4	5	7	6
11	11	12	13	15	14	11	11	12	13	15	14
10	10	8	9	11	10	10	10	8	9	11	10

$$\therefore A_3 = B_3$$

$$\therefore A_2 = \bar{B}_3 B_1 + B_3 \bar{B}_1$$

$$A_2 = B_3 \text{ XOR } B_1$$

		A_1						A_0			
$B_3 B_2$	$B_3 B_2$	00	01	11	10	$B_3 B_2$	$B_3 B_2$	00	01	11	10
		0	1	3	2			0	1	3	2
00	00	0	1	3	2	00	00	0	1	3	2
01	01	4	5	7	6	01	01	4	5	7	6
11	11	12	13	15	14	11	11	12	13	15	14
10	10	8	9	11	10	10	10	8	9	11	10

$$\therefore A_1 = B_2 \bar{B}_1 + \bar{B}_2 B_1$$

$$A_1 = B_2 \text{ XOR } B_1$$

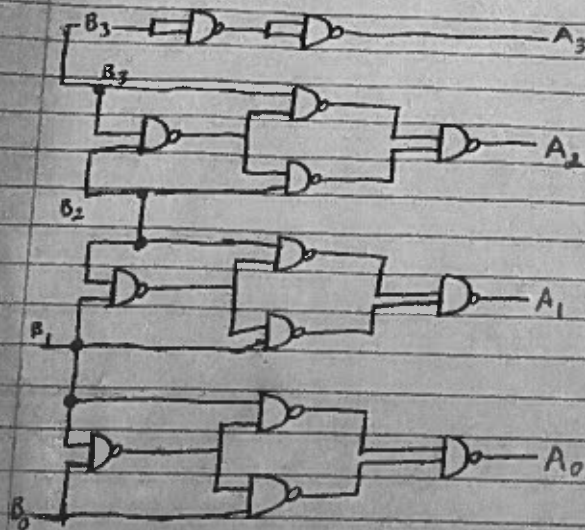
$$\therefore A_0 = \bar{B}_1 B_0 + B_1 \bar{B}_0$$

$$A_0 = B_1 \text{ XOR } B_0$$

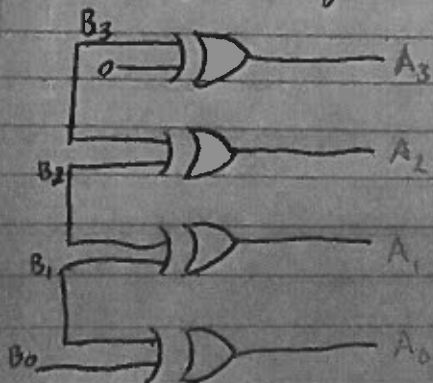
-Logic Circuits Implementations:-

• Logic Circuits Implementation:-

1) Using NAND gates -



2) Using XOR gates -



• Please discuss the following questions in your report. (a)What is the necessity of code conversions? (b)What is gray code? (c)What are universal gates? Why is it important?

(a)What is the necessity of code conversions?:

Code conversion is necessary at times when a user/system needs to use the output of one system as the input to another system. For it to operate properly, the conversion circuit has to be inserted in between both of the two systems when both of the systems use different codes for the same information. This is a great tool for machine language so they could effectively send information back and forth thus communicating with each other.

(b) What is gray code?:

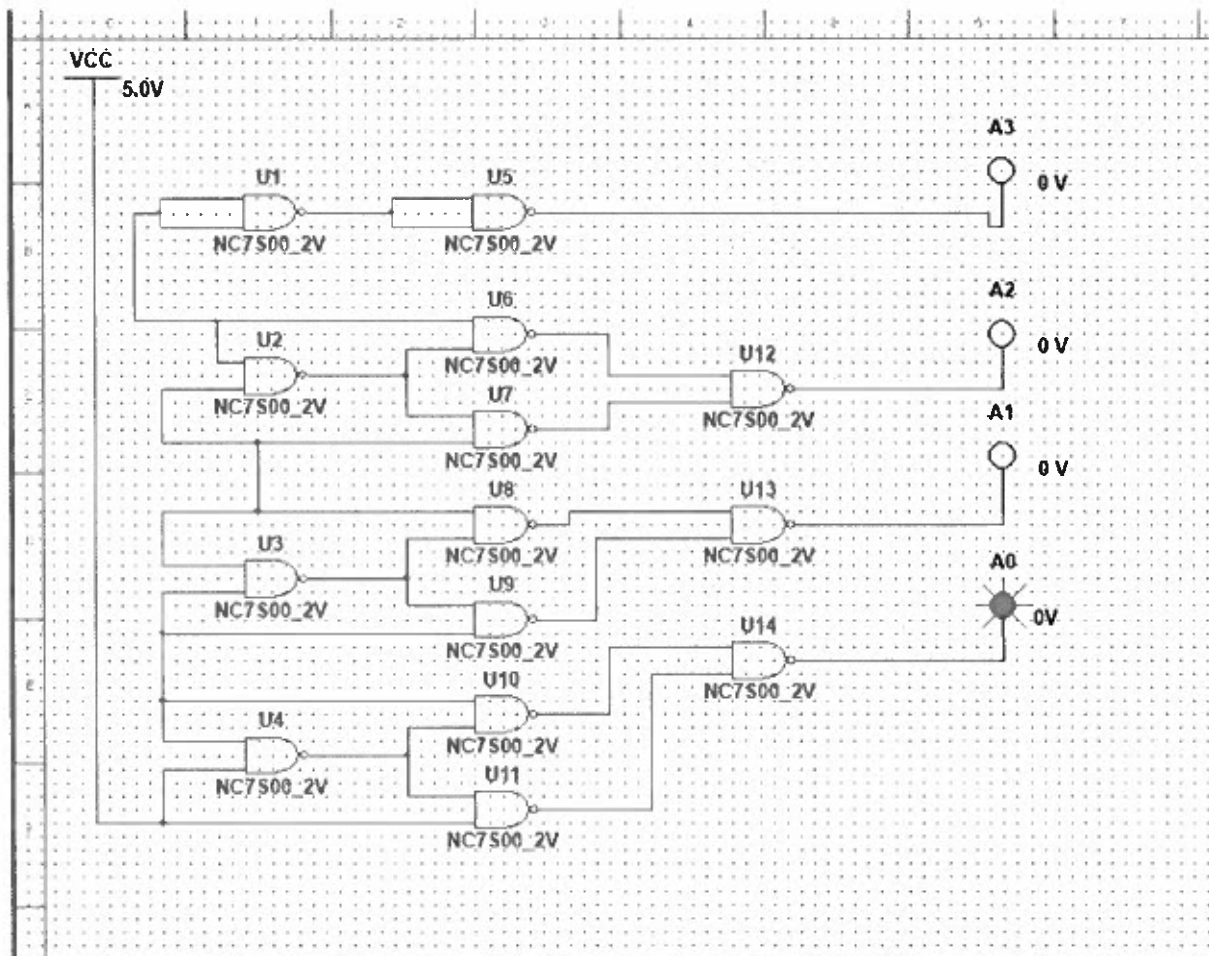
Gray code is a binary number system where all the successive pair of numbers differs in only one bit. They are very useful in the normal sequence of the binary numbers that are generated by the hardware that may cause some error during the transition from one number to the next number, because Gray code can easily eliminate this issue since only one bit actually changes its value during the transition between two numbers. Also, Gray code is not weighted which means it does not depend on a positional value of a digit.

© What are universal gates? Why is it important?

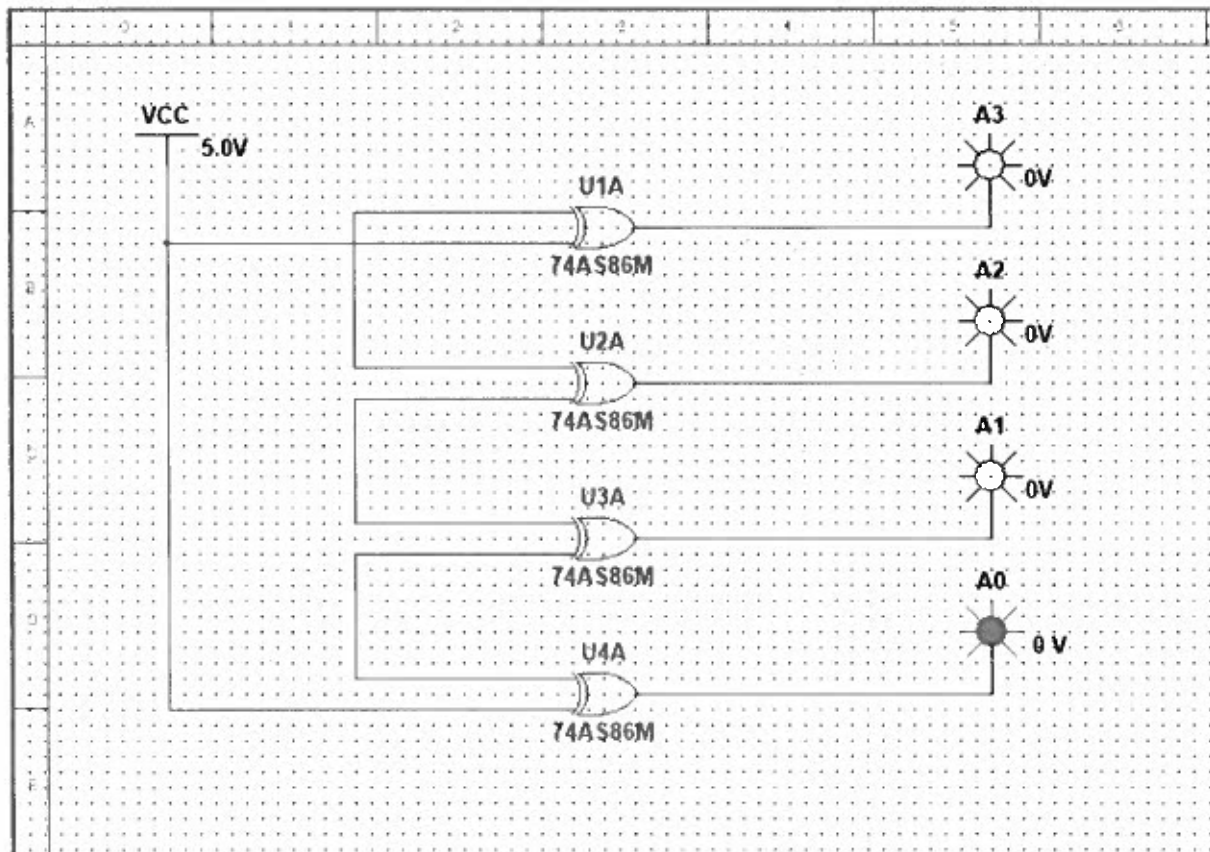
_____A universal gate is a gate that can implement any Boolean function without the need to use any other gate type. Also, universal gates can be used to construct all other logic gates. This is important for this project because NAND gates and XOR gates are considered universal gates, which are the two types of gates that we are working with in our logic implementations.

- If you could access the lab, you could physically implement the logic using appropriate ICs. If you could not access the lab, you could use Multisim to simulate the logic circuits. Please test whether your design is correct, and demonstrate the testing procedure in your report.

-Multisim logic implementation using only NAND gates:



-Multisim logic implementation using only XOR gates:



- **Conclusion:**

In conclusion, this project has given me a more in-depth understanding of universal gates and the importance of code conversion between binary and gray code. All of these things are very useful in machine language and the transformation of information in between systems so they can better operate/communicate with each other and remain efficient.

ELEG 4330-P01 Introduction to Digital Design
Fall 2021

Title of Assignment: PROJECT 2

Counter Design

ELEG 4330-P01 Introduction to Digital Design
Fall 2021

Project 2

Electrical and Computer Engineering DEPARTMENT
OUTCOMES SPECIFIC ASSIGNMENT COVER SHEET
ELEG 4330 & Introduction to Digital Design
FALL 2021

Instructor: Dr. Xiangfang Li

Title of Assignment

Counter Design

Outcome Measured Using this Assignment

Outcome 2

An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factor

Check the engineering design characteristics related to this assignment

X	Involves identifying the specific needs to be met by defining/understanding the problem and developing plans for the project to satisfy the need
X	Involves conducting a review of the literature and performing preliminary design by generating alternative ideas/solutions and applying creativity
X	Involves performing detailed engineering design leading to specification and fabrication/selection of components for system being designed.
	Involves consideration of one or more of the following (Please check those involved in this assignment.)
	☐ Public health
	☐ Safety and welfare
x	☒ Global factors
	☐ Cultural factors
	☐ Social factors
	☐ Environmental factors
x	☒ Economic factors
	Involves fabrication and testing of model or prototype

Brief justification of the suitability of assignment for the outcome

--

Prairie View A&M University
Department of Electrical and Computer Engineering
ELEG 4330 & Introduction to Digital Design

FALL 2021 (Due Dec. 2)

Project 2

In this project, **Design Binary Coded Decimal (BCD) sequence counters**, two parts will be performed. A project report is required.

In this project, sequential logic design will be performed. Both asynchronous and synchronous BCD counter design will be explored.

Part 1: BCD ripple counter design.

Part 2: Synchronous BCD counter design. Please follow the design procedure, construct a state diagram following the specification, construct state table, and choose the type of flip-flops to be used, then derive the simplified flip-flop input equations and draw the logic diagram.

Please implement both design in Multisim or HDL.

In your report, please describe the design process and the design results. Please test your design and show the test process and results in your report. In your conclusion, please describe your understanding of the differences of those two different counters.

Rubrics for Grading Design Project

ITEMS		POINTS ACHIEVED	MAXIMUM POINTS
1.	Introduction		10
2.	Demonstrate design process		30
3.	Your final logic circuits and simulation; Analysis and Interpretation of your design		30
4.	Discussion of Results and Conclusions		20
5.	General Write-up of report • Grammar and adequacy of plots and figures • Proper labeling and reference of graphs and tables • Clear and meaningful sentences		10
6.	Composite Report Grade		100
7.	Grade for student Outcome 2		50%

ABOVE AVERAGE



98

Course Number: ELEG 4330

Counter Design

Introduction to Digital Design

Submitted by:

Mohamed Ahmed

Date: 12/02/2021

Professor: Dr. Xiangfang

Table of Contents

1. Objective	1
2. Introduction	1
3. BCD ripple counter Methodology	1
3.1 State Diagram	1
3.2 Input Equations.....	2
3.3 Truth Table	2
3.4 Logic Circuit (BCD asynchronous Counter).....	3
4. Synchronous BCD counter Methodology.....	4
4.1 State Diagram	5
4.2 - State Table Synchronous BCD counter.....	5
4.3 - K-maps - BCD synchronous counter.....	7
4.4 Input Equations.....	9
4.5 Logic Circuit (BCD synchronous Counter).....	10
5. Testing counter.....	11
5.1 Testing of Asynchronous BCD counter.....	11
5.2 Testing of Synchronous BCD counter.....	16
6. Results	20
7. Conclusion	20
References	22

1. Objective

This project explores the sequential logic design procedures of the asynchronous BCD counter and the synchronous BCD counter.

2. Introduction:

There are two types of counters depending on which clock pulses are applied. The types are asynchronous counter and synchronous counter. In Asynchronous counter, also known as a Ripple Counter, different flip flops are triggered with different clock, not simultaneously. Synchronous Counter, on the other hands, all flip flops are triggered simultaneously with the same clock. During this project, we explore the design procedures of both the asynchronous BCD counter and synchronous BCD counter.

3. BCD ripple counter Methodology

BCD counter counts up to the number 10 with an applied clock signal. The BCD counter is 4-bit binary counter that counts from 0000 (0) to 1001 (9) in decimal form on the application of the clock signal. The logic diagram was derived in figure 3.2 from this knowledge. In this project, the BCD ripple counter was designed using four D flip-flops. A 4-bit counter counts from binary 0000 to binary 1111. A way to stop the count after ten was implemented using an AND gate to initiate a reset after 10. The logic diagram is shown in figure 3.3.

3.1 State Diagram

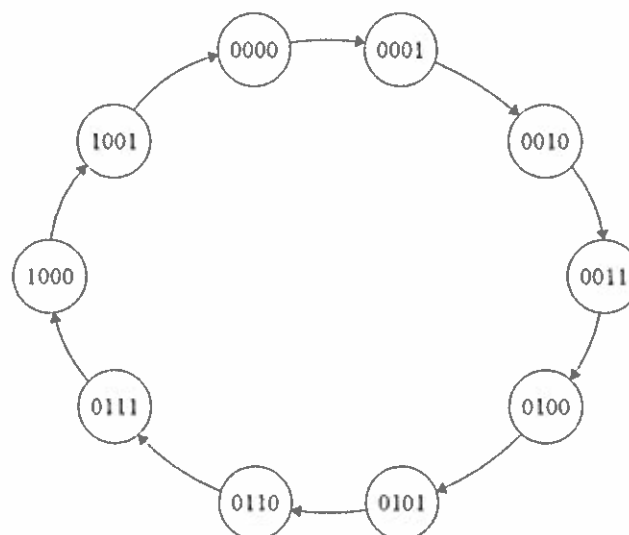


Figure 3.1 – BCD State Diagram

3.2 Input Equations

Input equation

$$D_0 = Q_0'$$

$$D_1 = Q_1'$$

$$D_2 = Q_2'$$

$$D_3 = Q_3'$$

$$R = Q_1Q_3$$

3.3 Truth Table

Decimal	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Table 3.1 – Truth Table for reset portion

3.4 Logic Circuit (BCD asynchronous Counter)

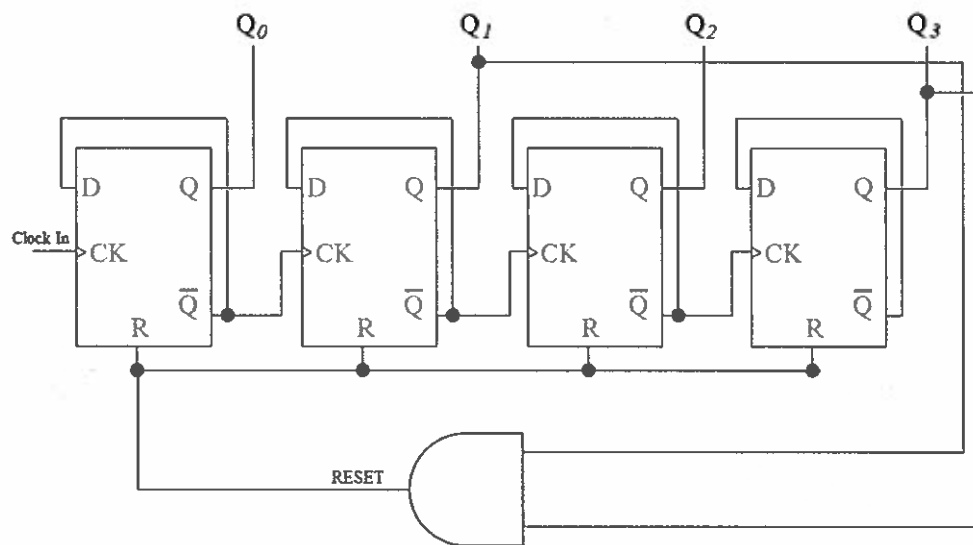


Figure 3.1 Logic Circuit Design (BCD asynchronous Counter – Count to 9)

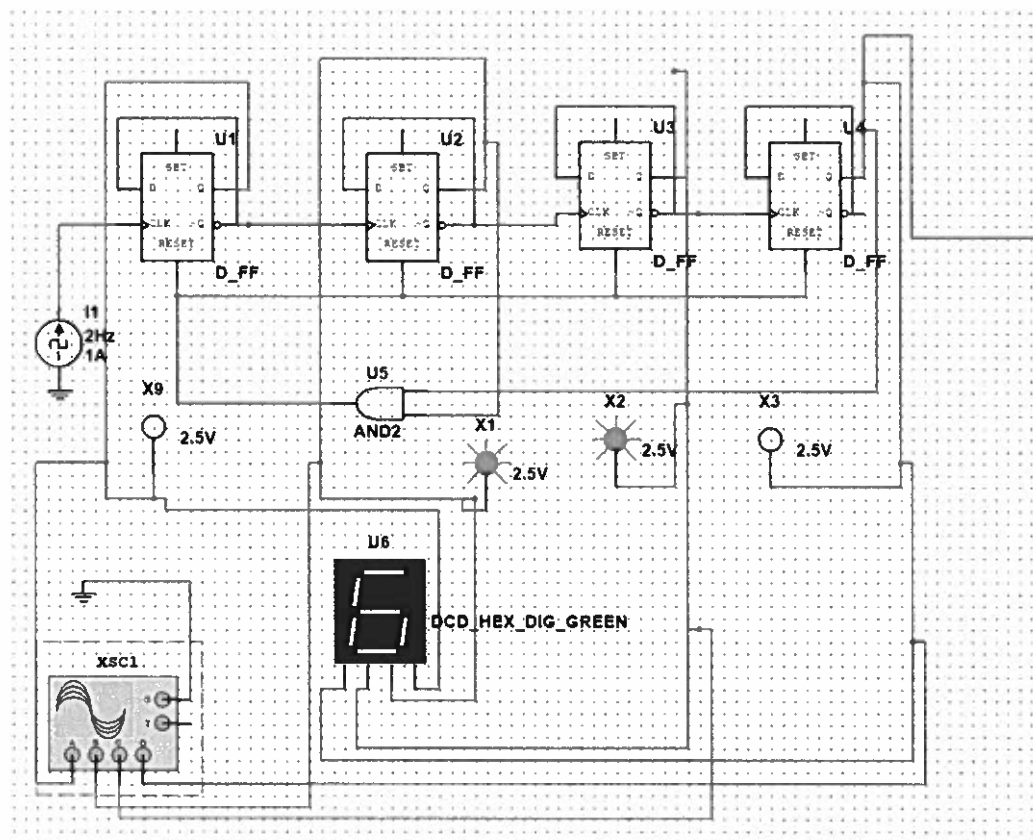


Figure 3.2 Logic Circuit (BCD asynchronous Counter – Count to 9) via Multisim

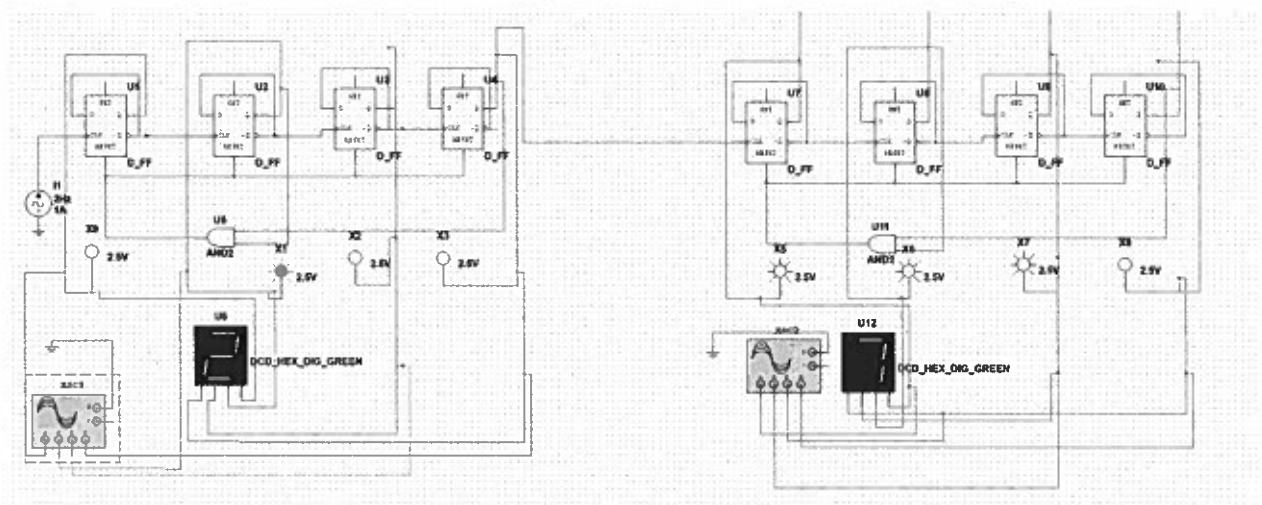


Figure 3.2 Logic Circuit (BCD asynchronous Counter – Count to 99)

4. Synchronous BCD counter Methodology

A BCD synchronous counter counts to the number 10 with an applied clock signal. The BCD counter is a 4-bit binary counter that counts from 0000 (0) to 1001 (9) in decimal form on the application of the clock signal. The BCD synchronous counter was designed using four JK flip-flops. A 4-bit counter counts from binary 0000 to binary 1111. A way to stop the count after ten was implemented using an AND gate to initiate a reset after 10. The State diagram is presented in figure 4.1. A JK-Flip Flop was chosen to be used for this design. The next step involved assigning binary values to the states and obtaining a state table shown in table 4.1 and table 4.2. A simplified flip-flop input equation was derived by using k-maps. The simplification procedure via k-maps is highlighted in figure 4.2 to figure 4.10. The logic diagram was drawn from the derived equations and is illustrated in figure 4.11. An extended counter that counts to 99 is shown in figure 4.12.

The design procedures followed to create a synchronous BCD counter are outlined below.

- A. Derived a state diagram for the circuit from the specifications given.
- B. Reduced the number of states if necessary.
- C. Assigned binary values to the states.
- D. Obtained the binary-coded state table.
- E. Chose the type of flip-flops to be used.
- F. Derived the simplified flip-flop input equations and output equations.
- G. Drew the logic diagram.

4.1 State Diagram

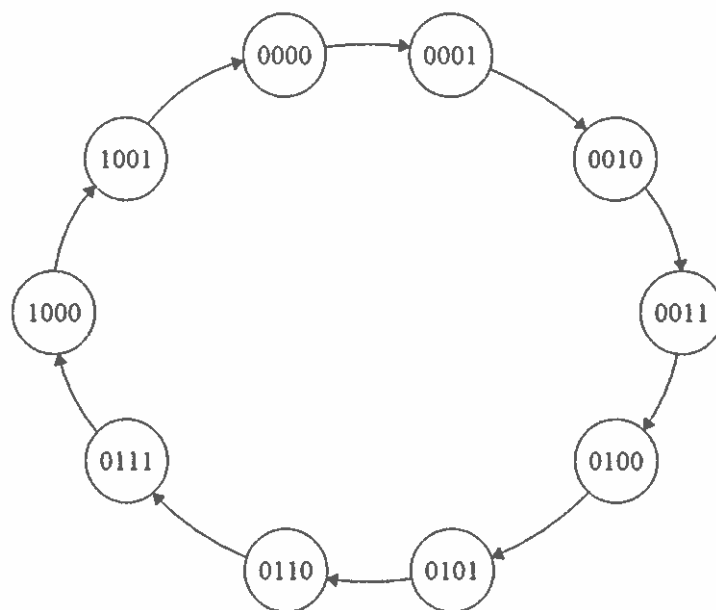


Figure 4.1 – BCD State Diagram

4.2 - State Table - Synchronous BCD counter

Decimal	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Table 4.1 – State Table for Present State (ABCD) portion

Decimal	A(t+1)	B(t+1)	c(t+1)	d(t+1)
0	0	0	0	1
1	0	0	1	0
2	0	0	1	1
3	0	1	0	0
4	0	1	0	1
5	0	1	1	0
6	0	1	1	1
7	1	0	0	0
8	1	0	0	1
9	0	0	0	0
10	x	x	x	x
11	x	x	x	x
12	x	x	x	x
13	x	x	x	x
14	x	x	x	x
15	x	x	x	x

Table 4.2 – State Table for Next State (ABCD) portion

Decimal	jaKa	JbKb	JcKc	JdKd
0	0x	0x	0x	1x
1	0x	0x	1x	x1
2	0x	0x	x0	1x
3	0x	1x	x1	x1
4	0x	x0	0x	1x
5	0x	x0	1x	x1
6	0x	x0	x0	1x
7	1x	x1	x1	x1
8	x0	0x	0x	1x
9	x1	0x	0x	x1
10	x	x	x	x
11	x	x	x	x
12	x	x	x	x
13	x	x	x	x
14	x	x	x	x
15	x	x	x	x

Table 4.3 – JK flip flop table

Decimal	A	B	C	D	Reset
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	x
12	1	1	0	0	x
13	1	1	0	1	x
14	1	1	1	0	x
15	1	1	1	1	x

Table 4.4 – Truth Table for reset portion

4.3 - K-maps - BCD synchronous counter

AB \ CD	00	01	11	10
00				
01			1	
11	x	x	x	x
10	x	x	x	x

Figure 4.2 Ja K-map

Ja = BCD

AB \ CD	00	01	11	10
00	x	x	x	x
01	x	x	x	x
11	x	x	x	x
10		1	x	x

Figure 4.3 Ka K-map

$$K_a = D$$

AB \ CD	00	01	11	10
00			1	
01	x	x	x	x
11	x	x	X	x
10			X	x

Figure 4.4 Jb K-map

$$J_b = CD$$

AB \ CD	00	01	11	10
00	x	x	x	x
01			1	
11	x	x	x	x
10	x	x	x	x

Figure 4.5 Kb K-map

$$K_b = CD$$

AB \ CD	00	01	11	10
00		1	x	x
01		1	x	x
11	x	x	x	x
10			x	x

Figure 4.6 Jc K-map

$$J_c = A'D$$

AB \ CD	00	01	11	10
00	x	x	1	
01	x	x	1	
11	x	x	x	x
10	x	x	x	x

Figure 4.7 Kc K-map

$$K_c = D$$

AB \ CD	00	01	11	10
00	1	x	x	1
01	1	x	x	1
11	1	x	x	1
10	1	x	x	1

Figure 4.8 Jd K-map

$$J_d = 1$$

AB \ CD	00	01	11	10
00	x	1	1	x
01	x	1	1	x
11	x	1	1	x
10	x	1	x	x

Figure 4.9 Kd K-map

$$K_d = 1$$

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	x	x	x	x
10	0	0	x	1

Figure 4.10 Reset K-map

$$R = AC$$

4.4 Input Equations

Simplified input equation for BCD synchronous counter:

$$\begin{aligned}
 J_a &= BCD & K_a &= D \\
 J_b &= CD & K_b &= CD \\
 J_c &= A'D & K_c &= D \\
 J_d &= 1 & K_d &= 1 \\
 R &= AC
 \end{aligned}$$

Figure 4.11 – Logic Diagram BCD synchronous counter to 9

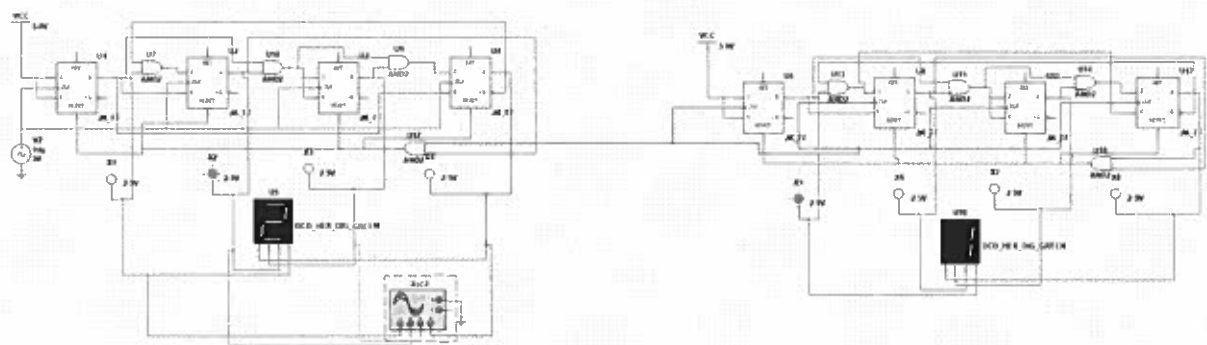


Figure 4.12 – Logic Diagram BCD synchronous counter to 99

5. Testing counter

Both BCD counters, asynchronous and synchronous, were tested via Multisim. Multisim testing results for the BCD asynchronous counter are shown in Figures 5.1 to 5.11. Figures 5.12 to 5.22 show testing results for BCD synchronous counter testing.

5.1 Testing of Asynchronous BCD counter

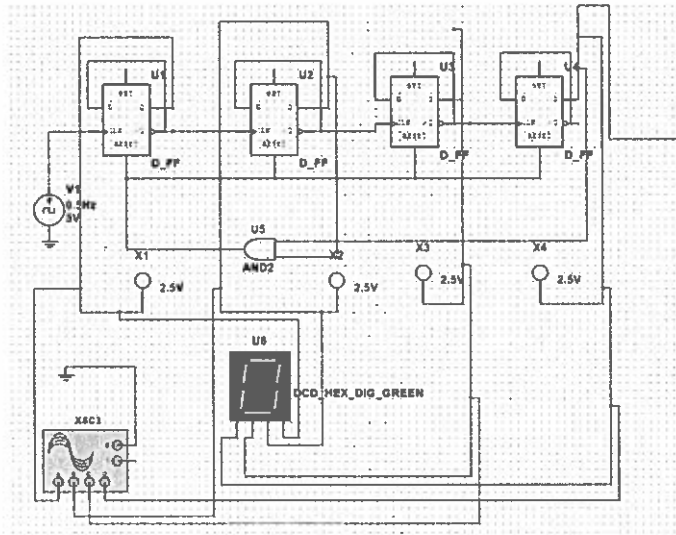


Figure 5.1 – BCD Asynchronous Testing (0000)

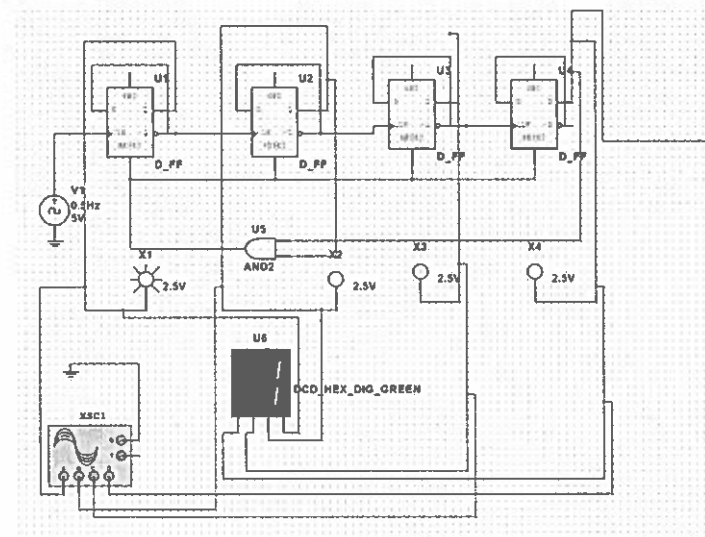


Figure 5.2 – BCD Asynchronous Testing (0001)

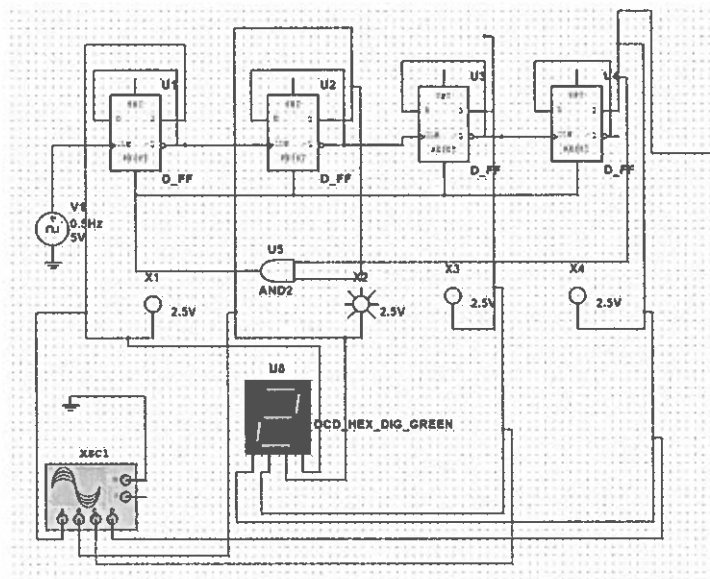


Figure 5.3 – BCD Asynchronous Testing (0010)

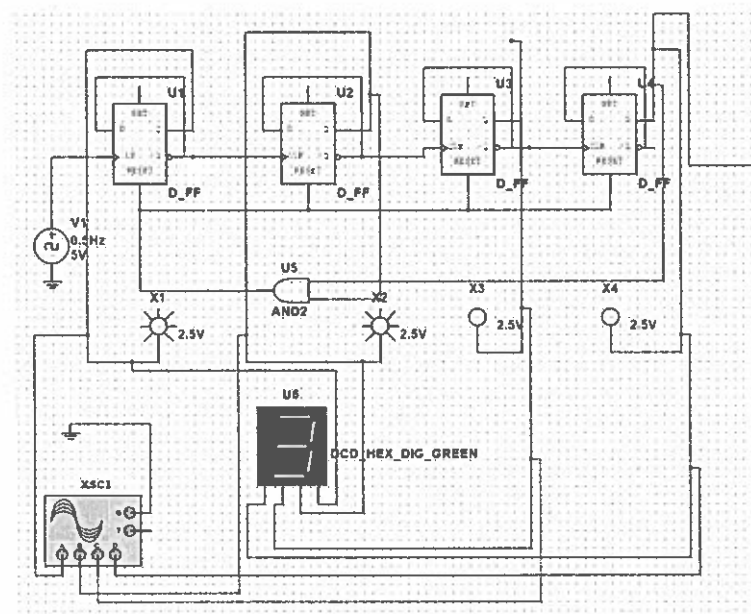


Figure 5.4 – BCD Asynchronous Testing (0011)

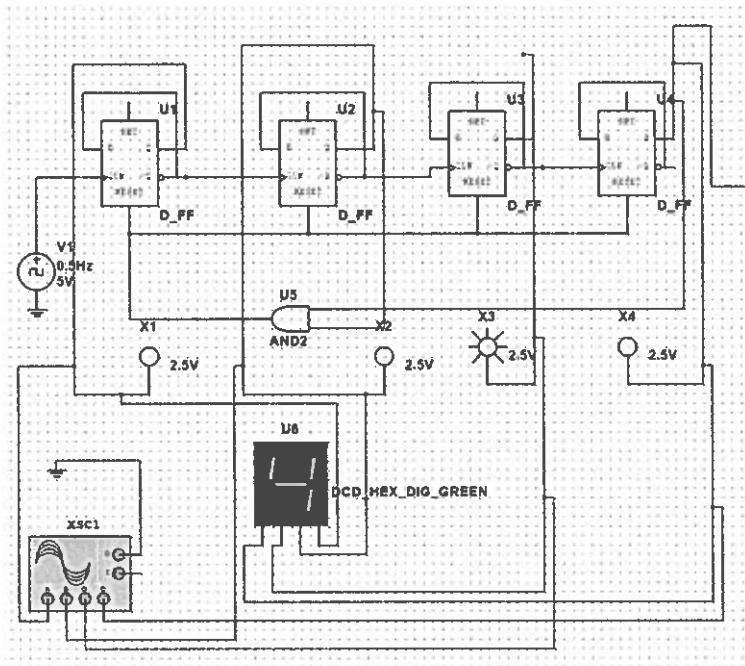


Figure 5.5 – BCD Asynchronous Testing (0100)

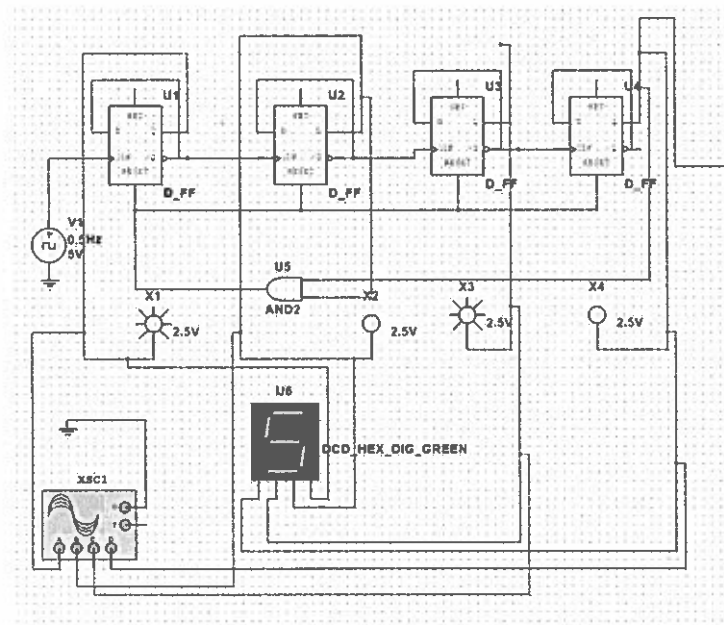


Figure 5.6 – BCD Asynchronous Testing (0101)

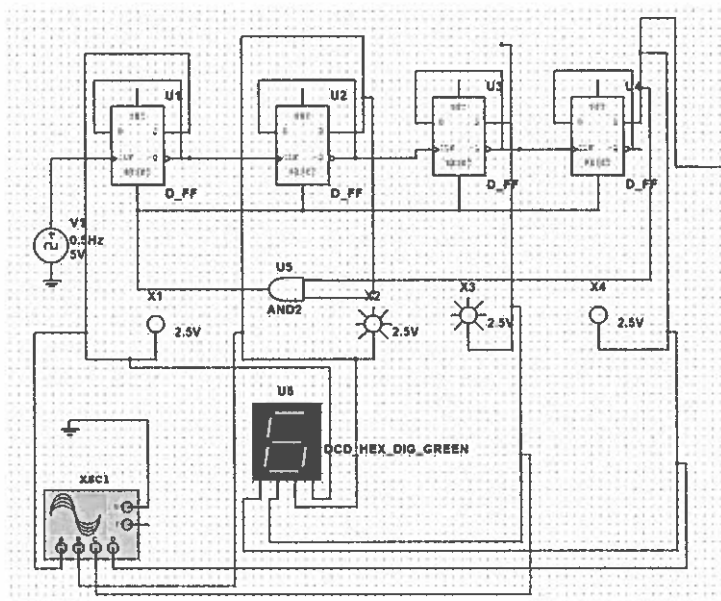


Figure 5.7 – BCD Asynchronous Testing (0110)

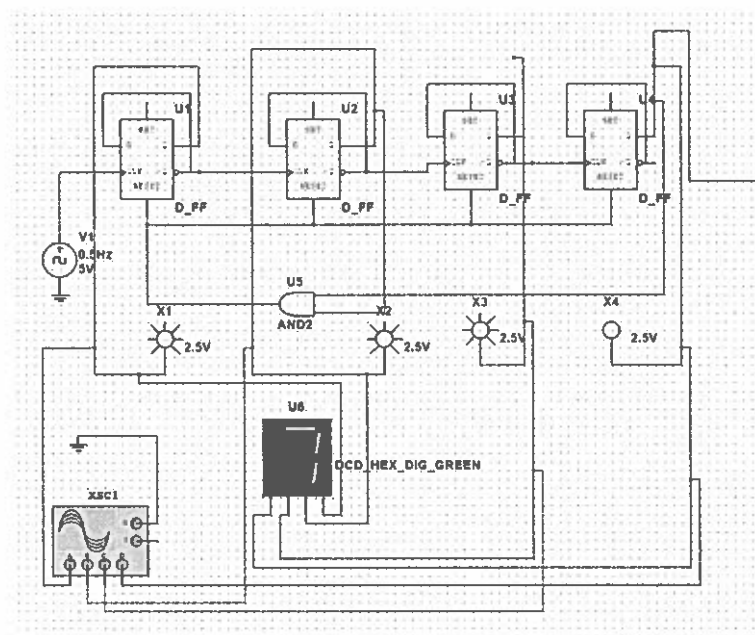


Figure 5.8 – BCD Asynchronous Testing (0111)

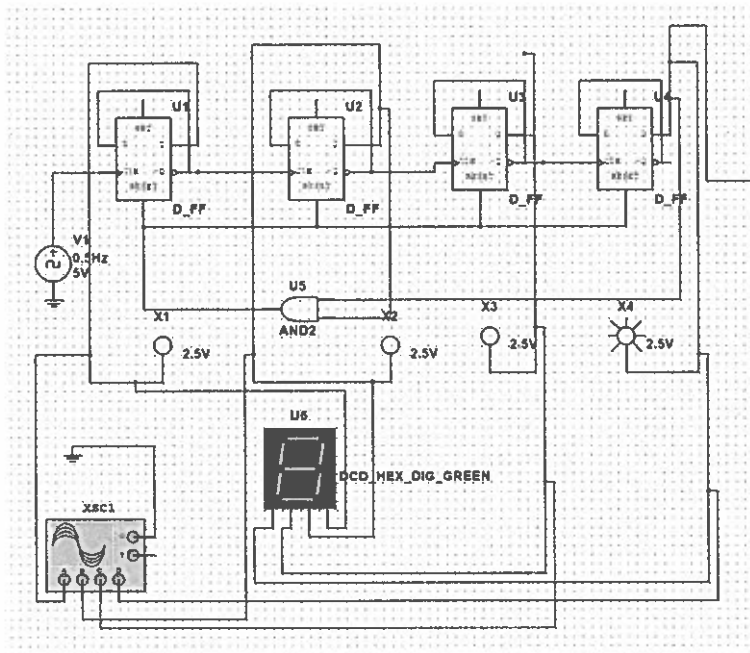


Figure 5.9 – BCD Asynchronous Testing (1000)

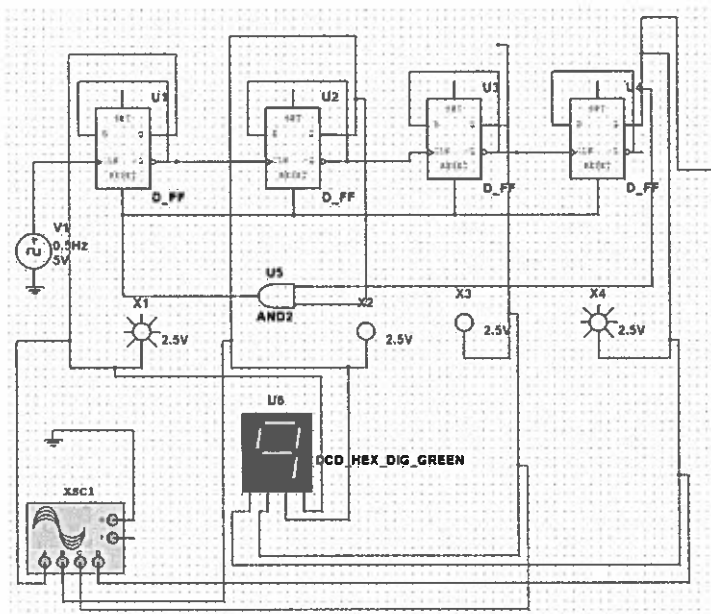


Figure 5.10 – BCD Asynchronous Testing (1001)

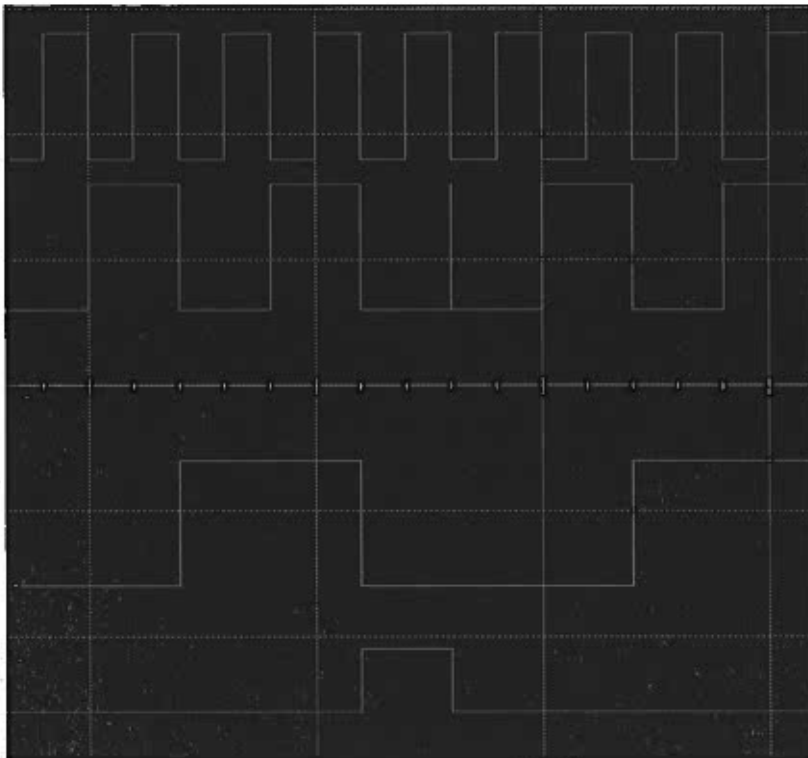


Figure 5.11 – BCD Asynchronous Timing Diagram

5.2 Testing of Synchronous BCD counter

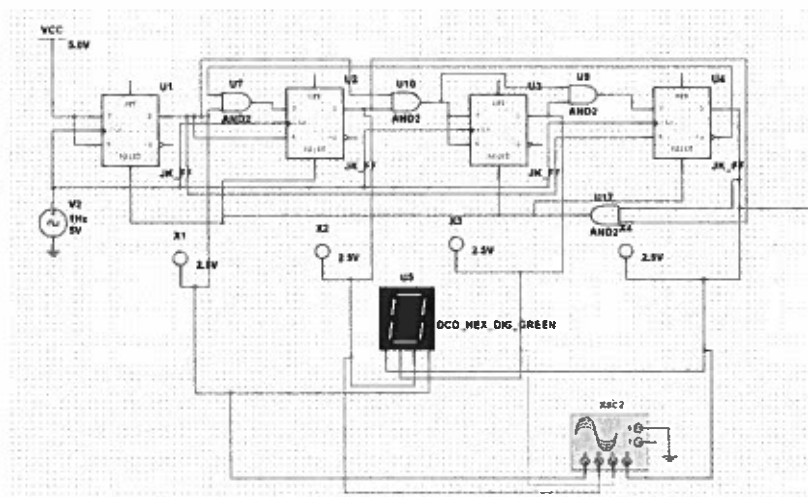


Figure 5.12 – BCD Synchronous Testing (0000)

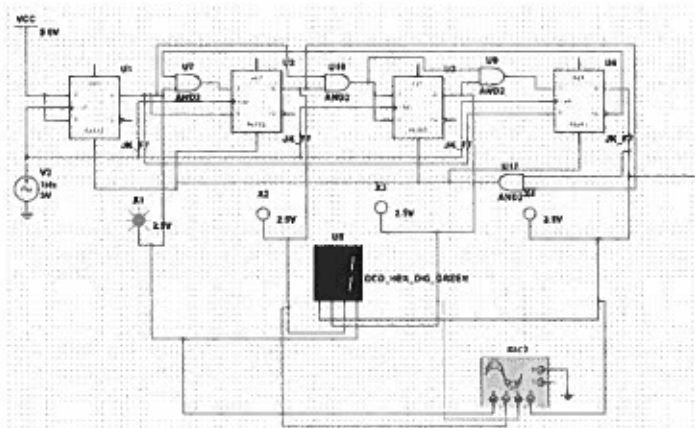


Figure 5.13 – BCD Synchronous Testing (0001)

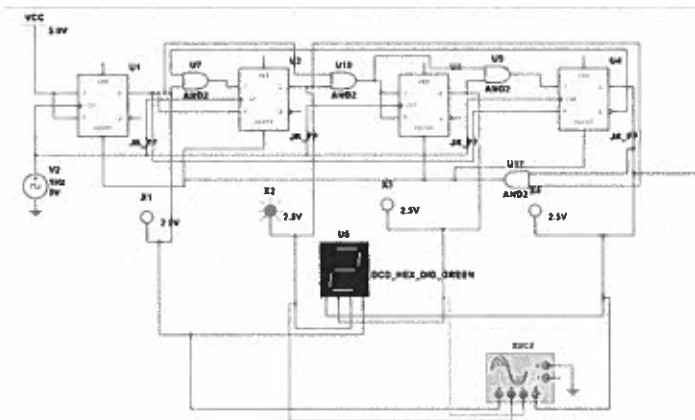


Figure 5.14 – BCD Synchronous Testing (0010)

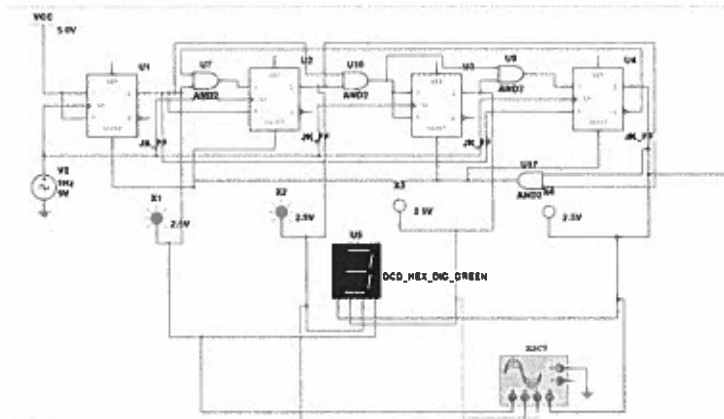


Figure 5.15 – BCD Synchronous Testing (0011)

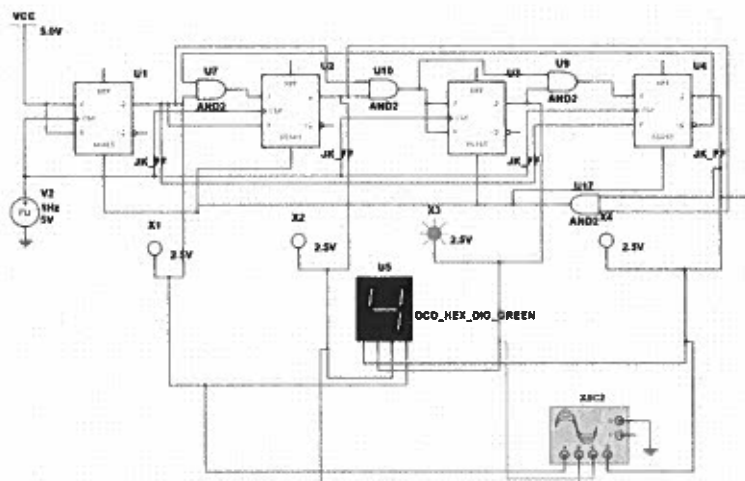


Figure 5.16 – BCD Synchronous Testing (0100)

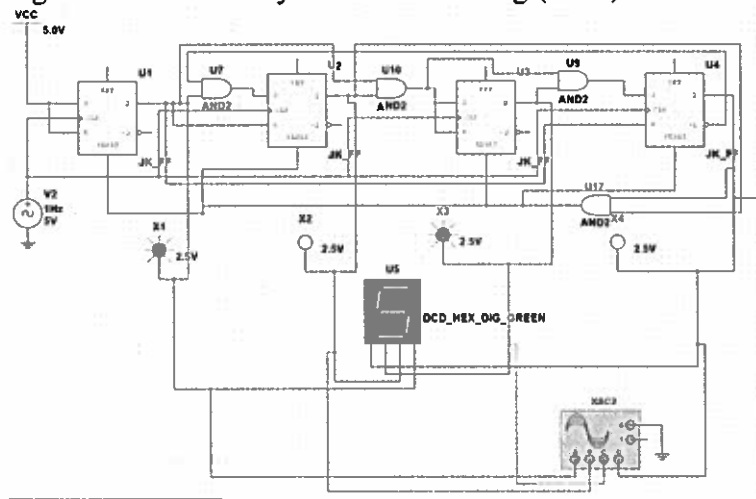


Figure 5.17 – BCD Synchronous Testing (0101)

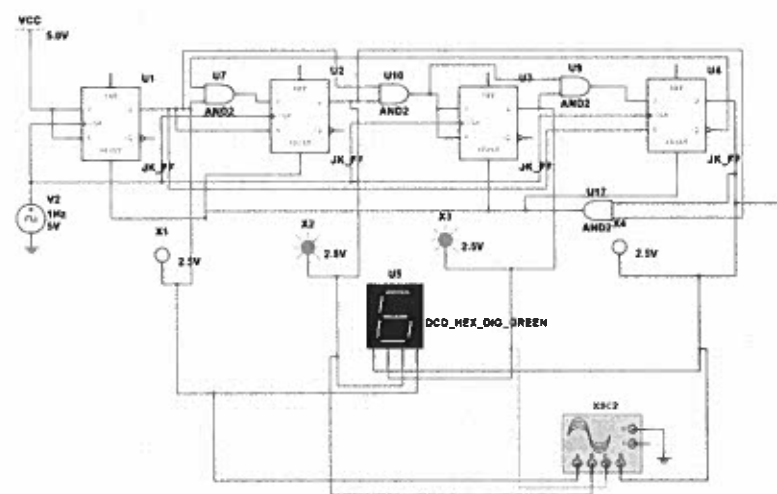


Figure 5.18 – BCD Synchronous Testing (0110)

Figure 5.21 – BCD Synchronous Testing (1001)

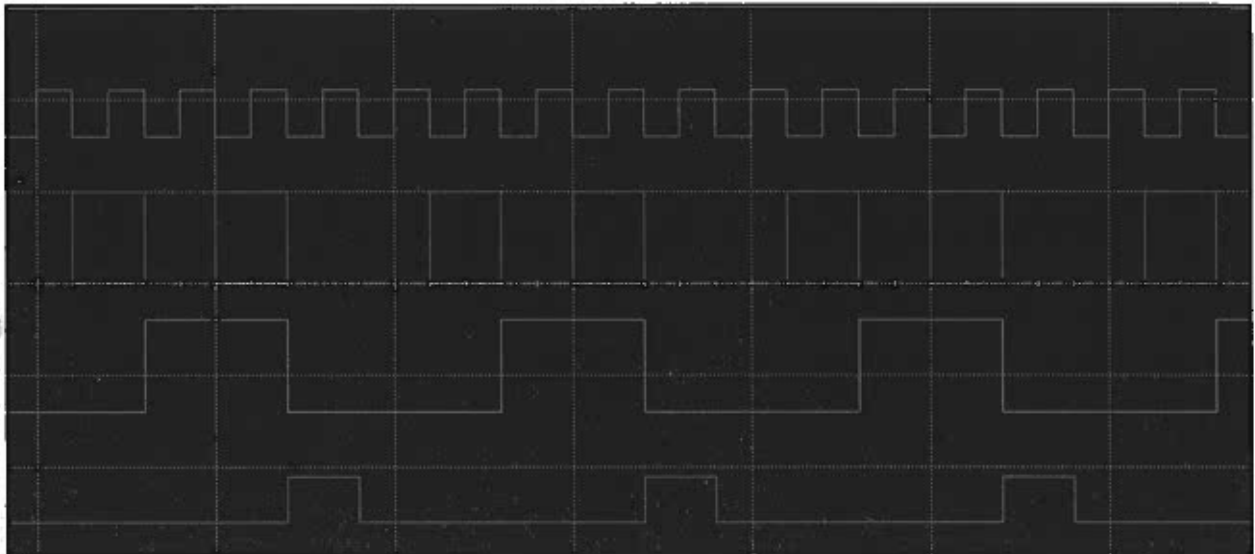


Figure 5.22 – BCD Synchronous Timing Diagram

6. Results

Both the asynchronous BCD counter and synchronous BCD counter ran without issue and counted from 0 to 9. There are some critical differences between the two counters. All flip-flops are triggered with the same clock simultaneously in the synchronous counter. While, in asynchronous counter, different flip flops are triggered with other clocks. Unlike an asynchronous counter, the state of the output bits changes simultaneously in the synchronous counter, without any ripple. This difference makes the synchronous counter faster than the asynchronous counter, leading to a high propagation delay. Since the clock signal in a synchronous counter is applied simultaneously to the clock inputs of all the flip-flops, there is no time lag between different outputs.

A major difference between the asynchronous and synchronous design was noted during the design procedure. Due to the asynchronous counter flip flops being triggered with different clocks, a systematic design approach could not be taken. A systematic methodology was followed in the synchronous design procedure, making it more design-friendly.

7. Conclusion

Counters are used in many applications. In places we come across the use of timers, we use counters of the synchronous type. Counters are used in digital electronics for counting reasons, they can count specific event happening in the circuit. Counters are especially designed synchronous sequential circuits, in which, the state of the counter is equivalent to the count assumed in the circuit by the flip flops. There are many applications of counters in sequential

circuits. Some applications are to count the number of occurrences, generate time sequences, and address memory.

There are differences between the asynchronous counter and the synchronous counter. In a synchronous counter, all flip flops are triggered with the same clock simultaneously. While, asynchronous counter, different flip flops are triggered with different clock. Unlike an asynchronous counter, the state of the output bits changes simultaneously in a synchronous counter, without any ripple. This difference leads to synchronous counter being faster in operation than asynchronous counter, leading to high propagation delay. Since the clock signal in a synchronous counter is applied to the clock inputs of all the flip-flops simultaneously, there is no time lag between different outputs.



References

- [1] M. M. Mano and M. D. Ciletti, *Digital design with an introduction to the Verilog HDL, VHDL, and systemverilog*. New York, NY: Pearson, 2019.

AVERAGE

88



Digital Design Project 1

Tri'Heem Neville

11/30/2021

Introduction

In this project, I had to design a binary-coded decimal (BCD) ripple counter and a synchronous BCD counter. The ripple counter can also be referred to as an asynchronous counter. To make these counters I had to use Flip-Flops, which are devices that are used as storage elements. To make the BCD ripple counter I decided to use D Flip-flops, the function of D Flip-Flops is output is equal to input. To make the synchronous BCD counter I used the T Flip-Flop, the function of the T Flip-Flop is to toggle the input to obtain the output. In order to complete these counters, I had to use the design process for sequential circuits. I will walk through the design process and how it translates into the Multisim simulation.

Design Process

The design process for sequential circuits begins with deriving the state diagram, the state diagram shows the behavior of the design. In this case, the behavior of the counters will count from zero to nine and then back to zero and repeat that sequence. The second step of the design process is creating the state table based on the state diagram. The state table shows the present state (p.s), next state (n.s) , inputs, outputs, and Flip-Flop inputs. For the BCD ripple counter the state table only has a present state, next state, and D flip-flop inputs. The state table for the synchronous BCD counter has present state, next state, output, and T Flip-Flops. The third step of the design process is to obtain the input equations from the state table using K-maps. The final step of the design process is drawing the circuit to verify that the design is correct. I did this using a Multisim simulation which will be outlined in the next section to show the correctness of the counters. The figures below represent the design process for the BCD ripple counter using D flip-flops.

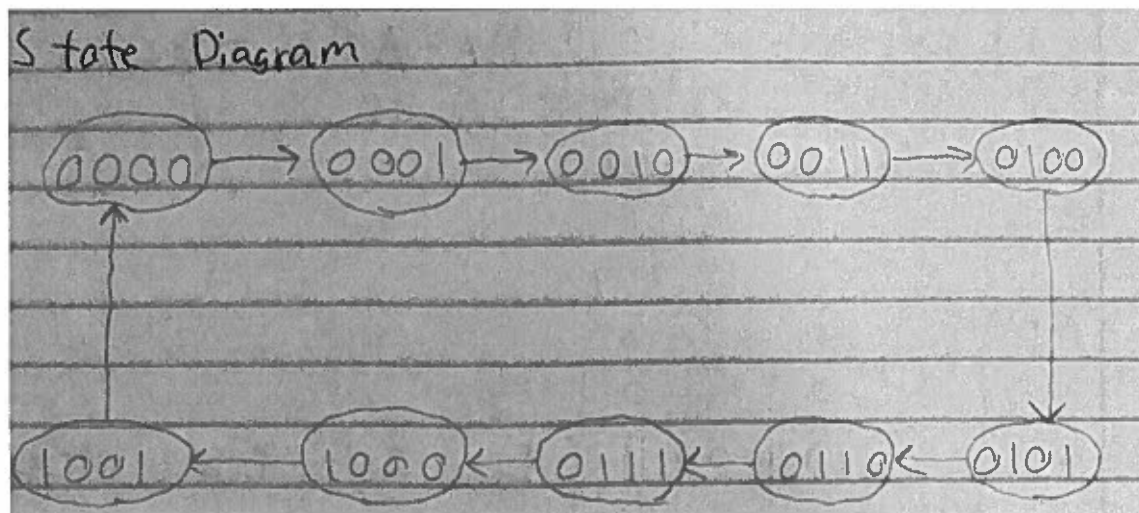


Figure 1.1 BCD Ripple Counter State Diagram

State Table											
P.S				N.S				Flip Flop Inputs			
A	B	C	D	A(t+1)	B(t+1)	C(t+1)	D(t+1)	D _A	D _B	D _C	D _D
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	0	0	0	0	0	0	0	0
1	0	1	0	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X

Figure 1.2 BCD Ripple Counter State Table

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	1	0	X	X

$$D_A = AC'D' + BCD$$

Figure 1.3 Input equation for D_A

AB \ CD	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	X	X	X	X
10	0	0	X	X

$$D_B = BC' + BD' + B'CD$$

Figure 1.4 Input equation for D_B

AB \ CD	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	X	X	X	X
10	0	0	X	X

$$D_C = A'C'D + A'CD'$$

Figure 1.5 Input equation for D_C

AB	CD				
		00	01	11	10
00		1	0	0	1
01		1	0	0	1
11		x	x	x	x
10		1	0	x	x

$D_0 = D'$

Figure 1.6 Input equation for D_0

The figures below represent the design process for the BCD synchronous counter using T flip-flops.

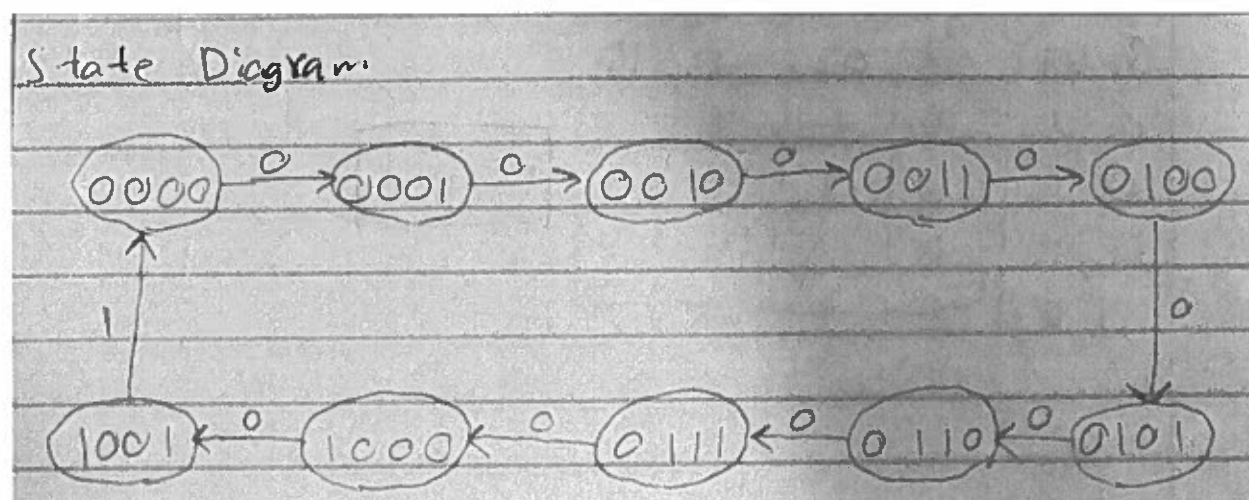


Figure 2.1 Synchronous BCD Counter State Diagram

State Table												
P.S				N.S				Output	Flip Flop Inputs			
Q ₈	Q ₄	Q ₂	Q ₁	Q ₈	Q ₄	Q ₂	Q ₁	Y	T _{Q₈}	T _{Q₄}	T _{Q₂}	T _{Q₁}
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	1	1	0	0	1
1	0	1	0	x	x	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x	x	x

Figure 2.2 Synchronous BCD Counter State Table

Q ₈ Q ₄ \ Q ₂ Q ₁		00	01	11	10
00	00	0	0	0	0
01	00	0	0	1	0
11	00	x	x	x	x
10	00	0	x	x	x

$$T_{Q8} = Q_8 Q_1 + Q_4 Q_2 Q_1$$

Figure 2.3 Input equation for T_{Q8}

$Q_8 Q_4$		$Q_2 Q_1$			
		00	01	11	10
0	0	0	0	1	0
0	1	0	0	1	0
1	1	x	x	x	x
1	0	0	0	x	x

$T_{Q4} = Q_2 Q_1$

Figure 2.4 Input equation for T_{Q4}

$Q_8 Q_4$		$Q_2 Q_1$			
		00	01	11	10
0	0	0	1	1	0
0	1	0	1	1	0
1	1	x	x	x	x
1	0	0	0	x	x

$T_{Q2} = Q_8' Q_1$

Figure 2.5 Input equation for T_{Q2}

$Q_8 Q_4$		$Q_2 Q_1$			
		00	01	11	10
0	0	1	1	1	1
0	1	1	1	1	1
1	1	x	x	x	x
1	0	1	1	x	x

$\overline{T_{Q1}} = 1$

Figure 2.6 Input equation for T_{Q1}

Multisim Simulation

After completing the design process, I was then able to build the circuits on Multisim to test their functionality which is the last step in the design process. To build the BCD ripple counter on Multisim, I needed a power source, a ground connection, seven AND gates, three OR gates, four D Flip-Flop, four probes, and a DCD_HEX display to see the numbers counting. In this section I will show that the simulation is functioning properly and can count any numbers from zero to nine. Below is a simulation of the BCD ripple counter using D Flip-Flops counting from five-seven as shown in the display on the right.

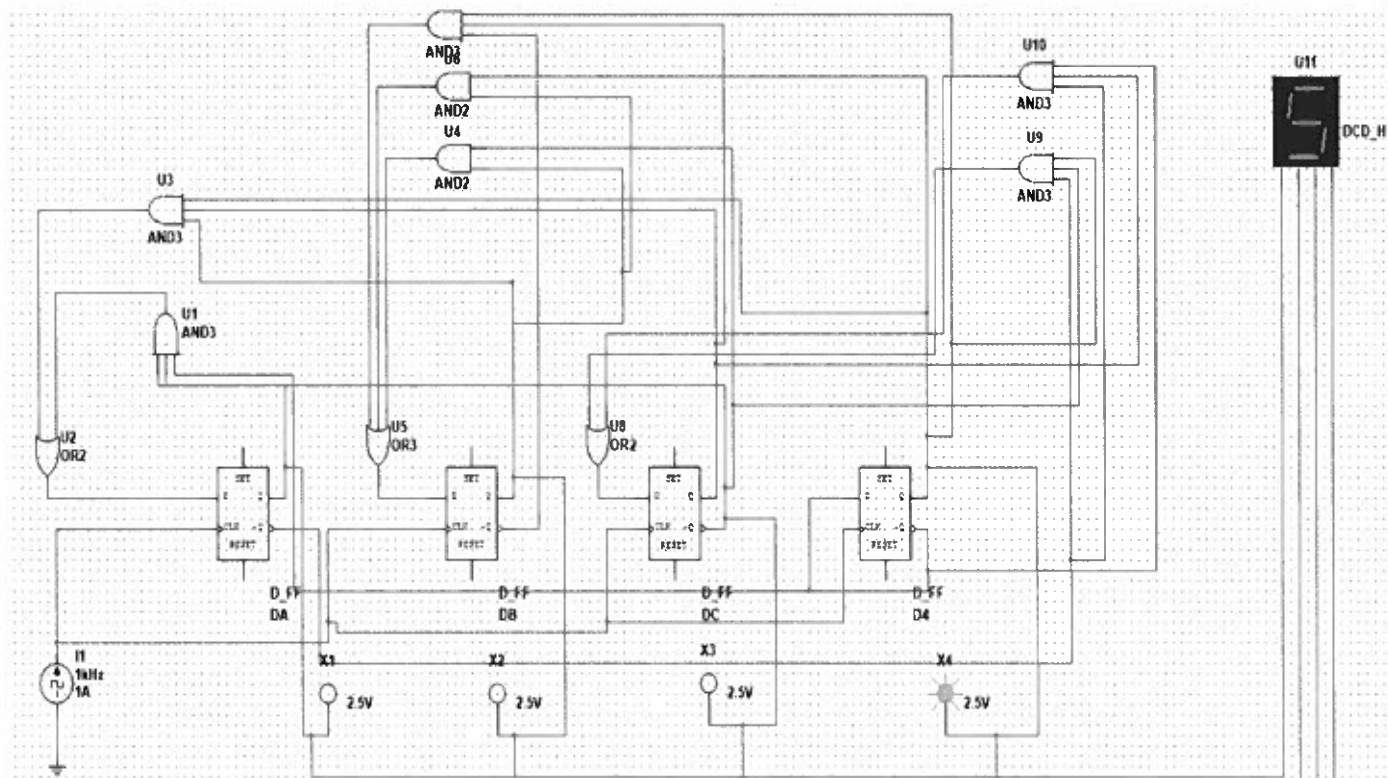


Figure 3.1 BCD Ripple Counter starting at five



To build the synchronous BCD counter on Multisim, I needed a power source, a ground connection, four AND gates, one OR gates, four T Flip-Flop, four probes, and a DCD_HEX display to see the numbers counting. Below is a simulation of the synchronous BCD counter using T Flip-Flops counting from 3-5 as shown in the display on the right.

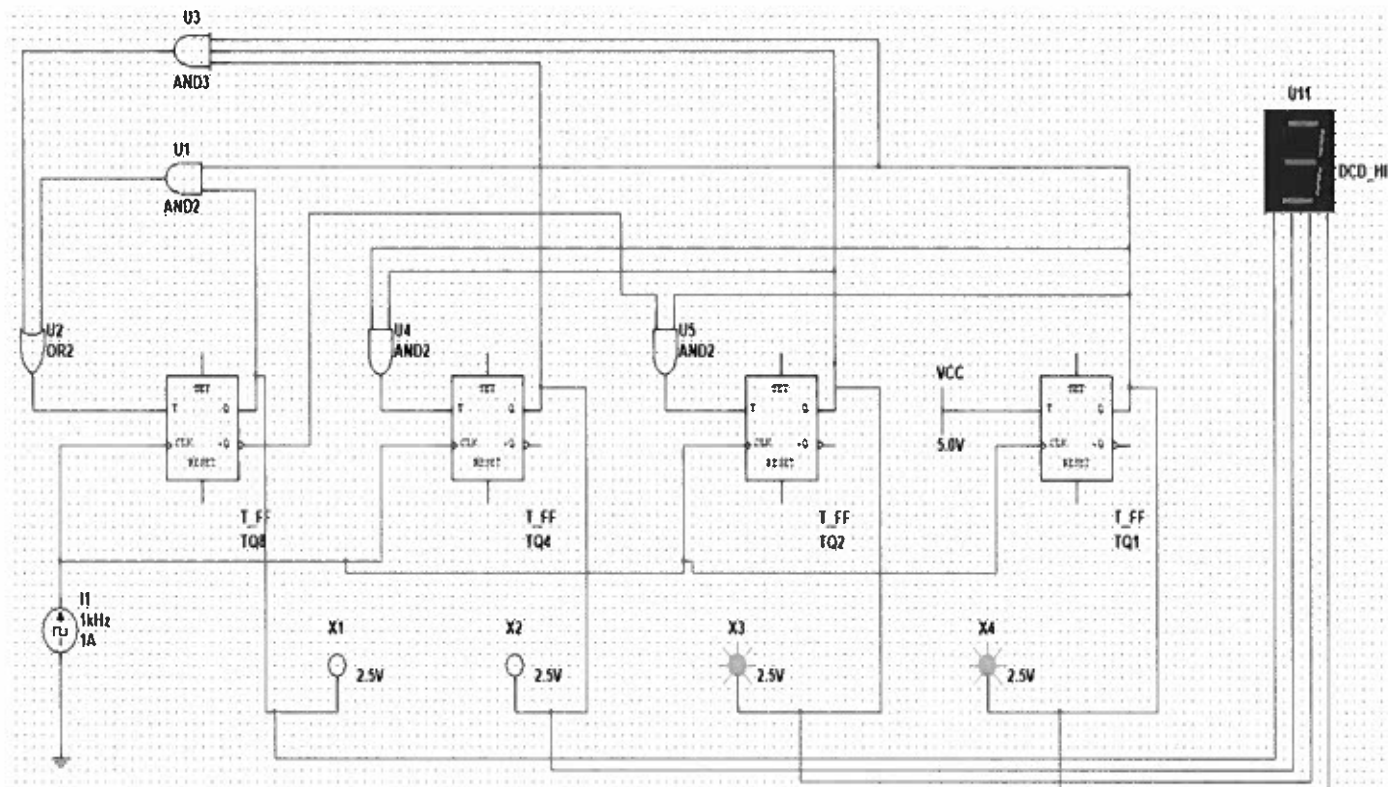


Figure 4.1 Synchronous BCD Counter starting at three

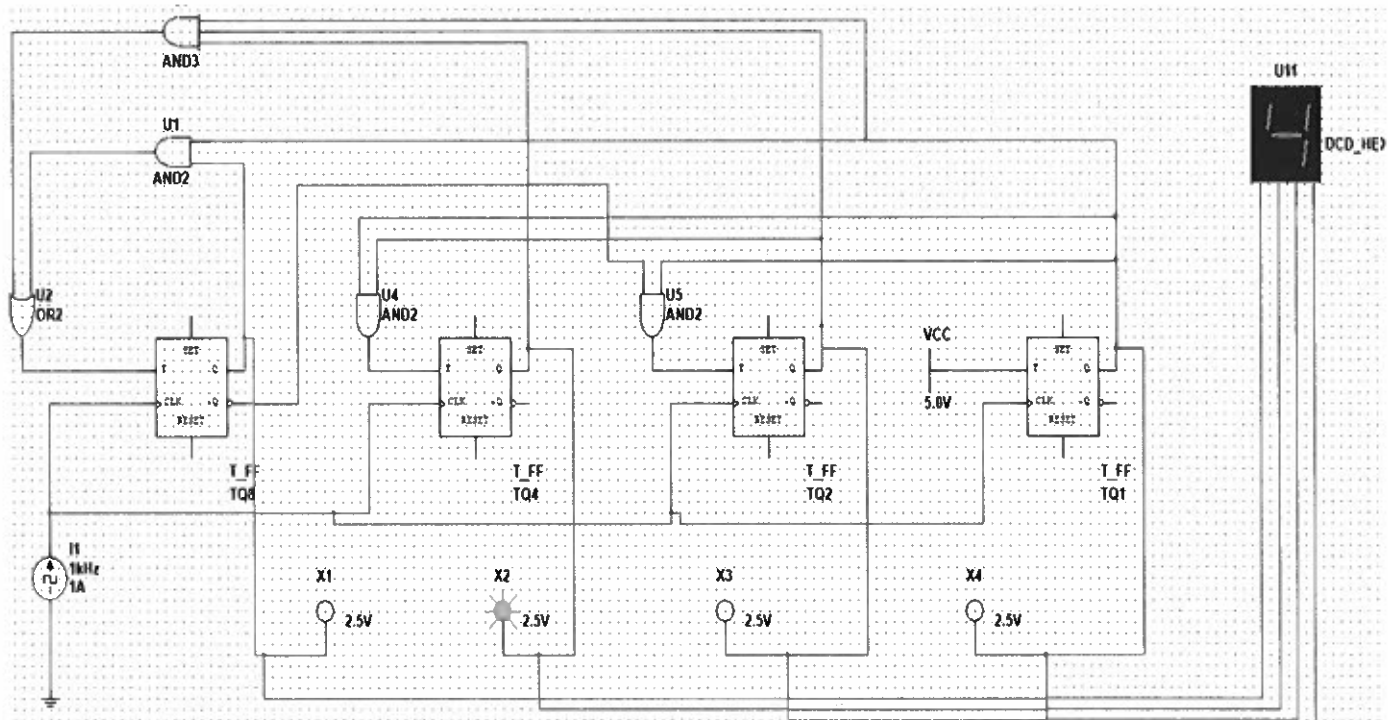


Figure 4.2 Synchronous BCD Counter changing to four

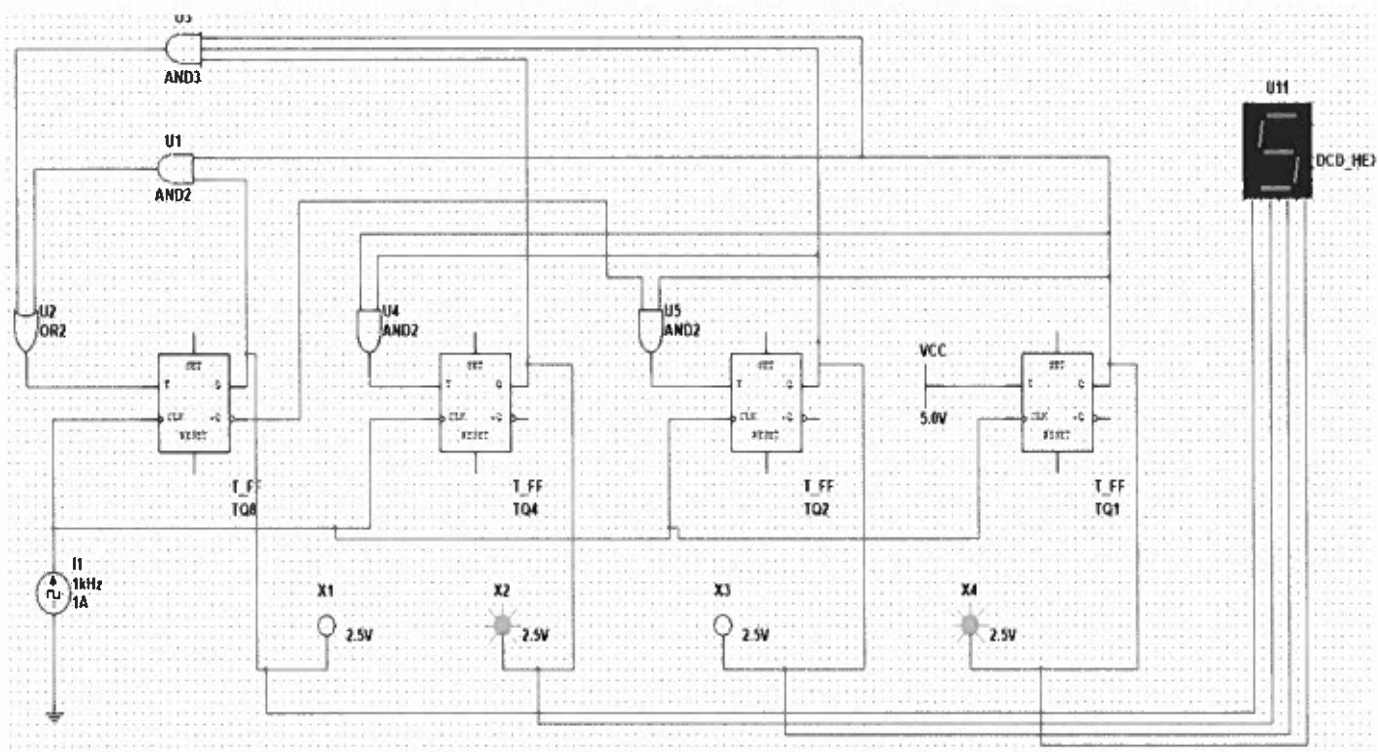


Figure 4.3 Synchronous BCD Counter changing to five

Conclusions

In conclusion the project was completed successfully, and the counter circuits are functioning as intended as shown in the Multisim simulation. The reason the counters only count up to nine from zero and the other six numbers are "don't cares", is because binary-coded decimal only goes up to nine and nothing higher. The main the difference between an asynchronous counter and a synchronous counter, is that the flip flops in the asynchronous counter have different clocks that are not triggered simultaneously. The flip-flops in the synchronous counter are only triggered by the same clock simultaneously which makes it faster. Another key difference between the two designs is that the asynchronous counter is also referred to as the serial counter. The synchronous counter is also referred to as the parallel counter.

BELOW AVERAGE

PRAIRIE VIEW A&M UNIVERSITY
ROY G. PERRY COLLEGE OF ENGINEERING

78



Course Number: ELEG 4330 Intro to Digital Design

Project 2: Sequential Logic Design Synchronous and Asynchronous

Submitted by *Marvin Berrios*

Date: 12/02/2021

Instructor: *Dr. Li*

TABLE OF CONTENTS

	Page
TABLE OF CONTENTS	2
BACKGROUND INFORMATION.....	3
OBJECTIVE	3
PROCEDURE & RESULTS.....	4
CONCLUSION	8

Background Information:

Sequential Logic is a form of binary circuit design that employs one or more inputs and one or more outputs, those whose states are related and define by rules that depend on previous states. Inputs and outputs can be either high or low, 1 or 0. This type of logic is used to construct finite state machines, which are the basic building block in all digital circuitry and in memory. All practical digital devices are a combination of combinational and sequential logic.

Types of sequential logic include synchronous and asynchronous sequential circuits.

Synchronous sequential logic circuits are digital sequential circuits in which the feedback to the input for next output generation is governed by clock signals. Asynchronous logic circuits are clockless or have some sort of timer. They use signals that indicate completion of operations, specified by simple data protocols.

A Binary Coded Decimal (BCD) counter is a series type of digital counter which is designed to count ten digits, numbers 0-9. It performs the operation of resetting automatically when there is a new clock input signal. In my project I will create both a synchronous and asynchronous BCD counter to show the differences between types.

Objective:

To implement and design both asynchronous and synchronous BCD counter designs, specifically the BCD ripple counter design and a synchronous BCD counter design. Additionally, to show the ability to apply engineering design to produce solutions that meet specified needs with consideration of safety and public health.

Procedure & Results:

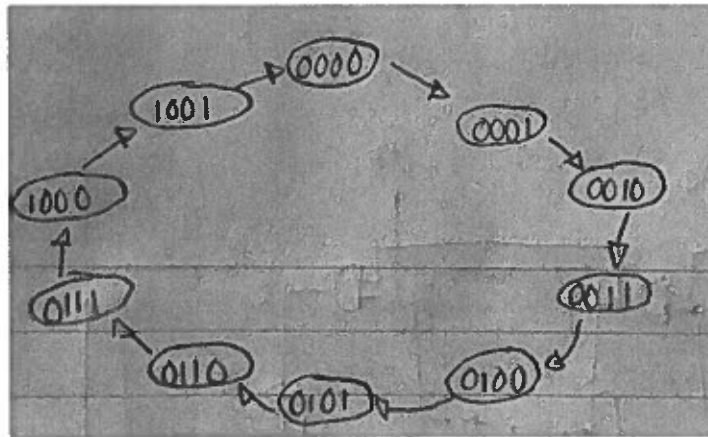


Figure 1: Counter State Diagram

Figure one above shows the state diagram of a BCD counter, the direction and order in which it is set up can account for the overall design of both the synchronous and asynchronous models.

Part 1: BCD Ripple counter Design

PS				NS			
Q ₃	Q ₄	Q ₂	Q ₁	Q ₃	Q ₄	Q ₂	Q ₁
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0

Figure 2: State Table Part 1

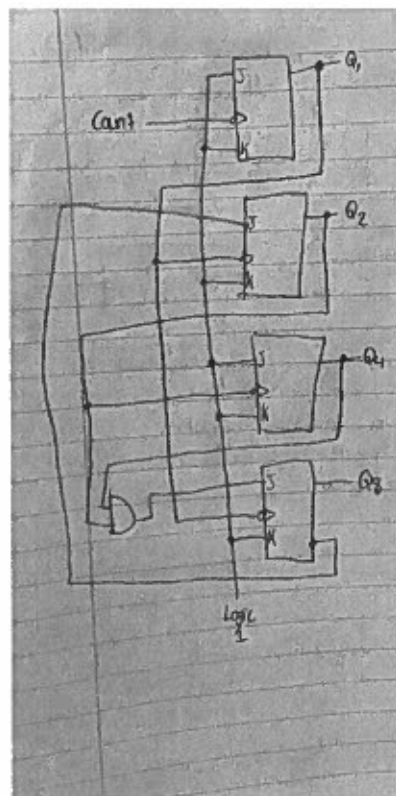


Figure 3: Logic Circuit Part 1

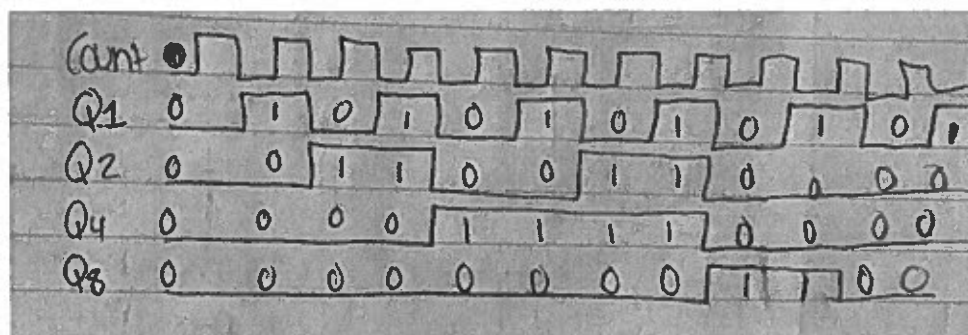


Figure 4: BCD Ripple output

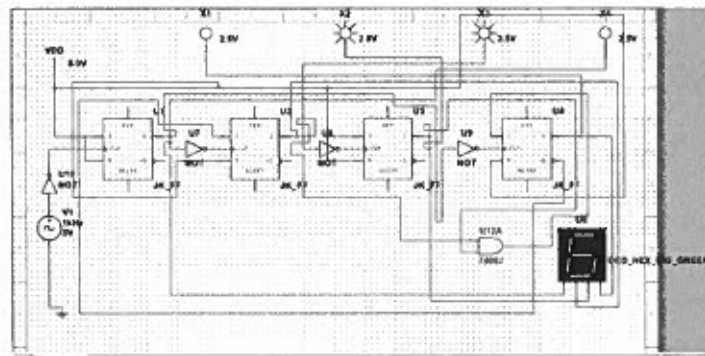


Figure 5: Simulink Part 1

Part 2: Synchronous BCD Counter Design

PS				MS				FF Inputs							
Q_3, Q_2, Q_1, Q_0				Q_3, Q_2, Q_1, Q_0				$J_3, K_3, J_2, K_2, J_1, K_1, J_0, K_0$							
0 0 0 0				0 0 0 1				0 X 0 X 0 X X X							
0 0 0 1				0 0 1 0				0 X 0 X 1 X X 1							
0 0 1 0				0 0 1 1				0 X 0 X X 0 1 X							
0 0 1 1				0 1 0 0				0 X 1 0 X 1 X 1							
0 1 0 0				0 1 0 1				0 X X 0 0 X 1 X							
0 1 0 1				0 1 1 0				0 X X 0 1 X X 1							
0 1 1 0				0 1 1 1				0 X X 1 X 0 1 X							
0 1 1 1				1 0 0 0				1 X X X X 1 X 1							
1 0 0 0				1 0 0 1				X 1 0 X 0 X X 1							
1 0 0 1				0 0 0 0				X X X X X X X X							
1 0 1 0				X X X X				X X X X X X X X							
1 0 1 1				X X X X				X X X X X X X X							
1 1 0 0				X X X X				X X X X X X X X							
1 1 0 1				X X X X				X X X X X X X X							
1 1 1 0				X X X X				X X X X X X X X							
1 1 1 1				X X X X				X X X X X X X X							

$J_0 = K_0 = 1, J_1 = Q_3'Q_0, K_1 = Q_0, K_2 = Q_1Q_0', J_2 = Q_1Q_0, K_2 = Q_1'$
 $J_3 = Q_1Q_1Q_0$

Figure 6: State Table & Input Equations Part 2

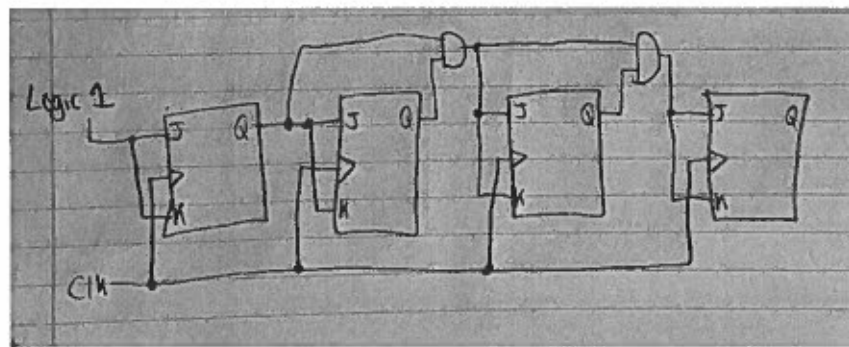


Figure 7: Logic Circuit Part 2

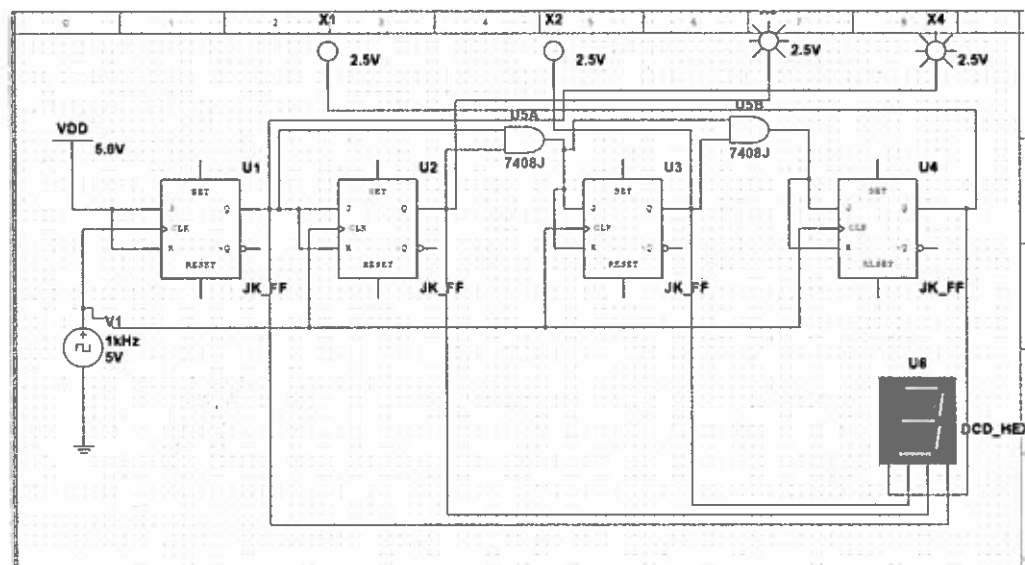


Figure 8: Multisim Part 2

As shown in Figure 8, the way I designed my circuit allows the output to change on the LCD hexadecimal display. So as soon as the circuit is running, the BCD starts counting from 0-9 repeatedly. This was done in both Simulink designs as well as the inclusion of the probes at the top of the designs which show the binary equivalent (from left to right) of the decimal numbers shown in the LED display. One of the main differences between part 1 and part 2 is the way the clock is used. In the synchronous model the clock is required throughout, but for the asynchronous it is not. I used the same flip flops so the differences could be easily distinguished. BCD counters have a sequence of 10 states, 0-9, and because of those 10 states, 4 flip flops are used.

Conclusion:

Throughout this project the understanding of both synchronous and asynchronous sequential logic was shown, and the understanding and implementation of BCD counters. For each individual part of the project, JK flip flops were used, the differences and similarities between asynchronous and synchronous designs can be seen in the results. The design procedure was followed and successfully completed. This was done with engineering application to meet the specified requirements and all things were considered.

ELEG 4330-P01 Introduction to Digital Design

Fall 2021

Title of Assignment: Final exam

TOPICS COVERED IN Final exam

Test 1 covered the following topics:

1. Sequential digital logic; synchronous sequential logic vs Asynchronous sequential logic; Flip-flop; Latches
2. Analysis of clocked sequential logic circuits; state equation; state table; state diagram; Mealy and Moore state machine
3. Design of sequential logic circuit.
4. Register and counters
5. Memory and Programmable Logic

ELEG 4330-P01 Introduction to Digital Design
Fall 2021

Final exam

ELEG 4330 Introduction to Digital Design
Department of Electrical & Computer Engineering
PVAMU

Final Exam

Fall 2021

Name: _____

1	24	
2	20	
3	20	
4	10	
5	14	
6	12	
7	15	
Total	100+15	

Instructions:

- (1) This exam is close book and notes. One cheatsheet is allowed.
- (2) Cell phone is not allowed.
- (3) Please keep your middle steps in the exam paper. Only the final result will NOT give you full credit.

You have 120 minutes to complete the exam. The points for each problem are indicated. This exam counts for 25% of your course grade.

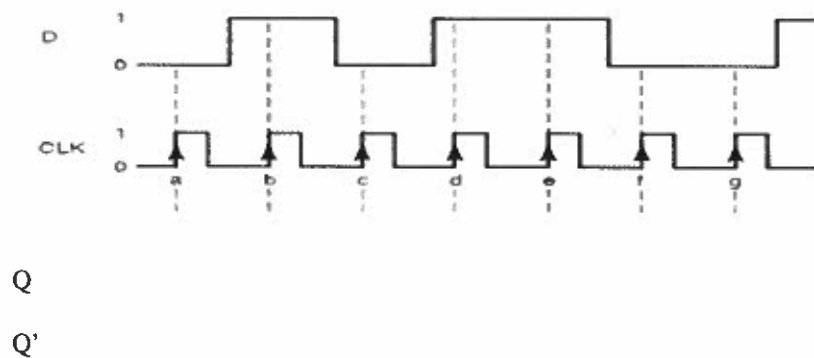
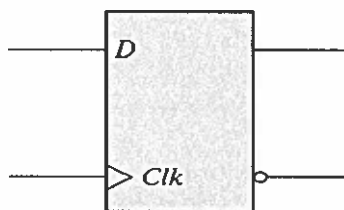
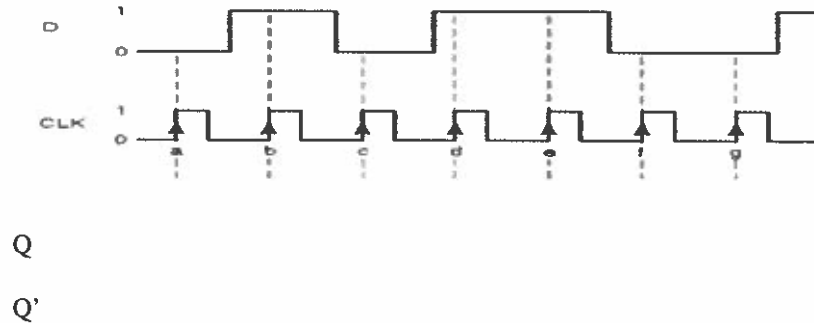
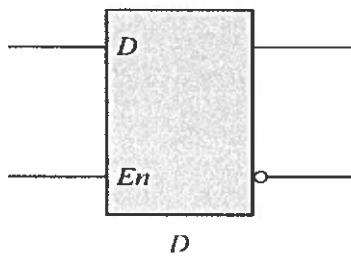
(24 pts) 1. Sequential digital logic

(4 pts) (1.1). Please briefly compare Combinational logic and Sequential logic.

(4 pts) (1.2). Please briefly compare Synchronous sequential logic circuits and Asynchronous sequential logic circuits.

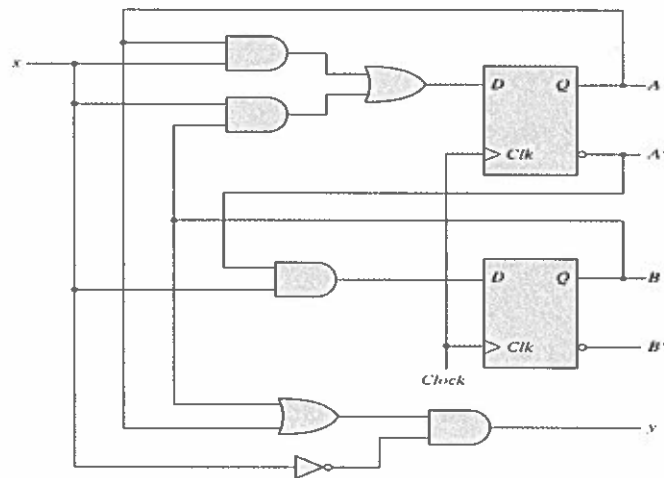
(6 pts) (1.3). Please demonstrate how to construct a **D flip-flop** from **D Latches** (with falling edge triggered).

(10pts) (1.4) . Given the following timing diagram, please draw the corresponding output(s) in the timing diagram. Assume Q start with 0.

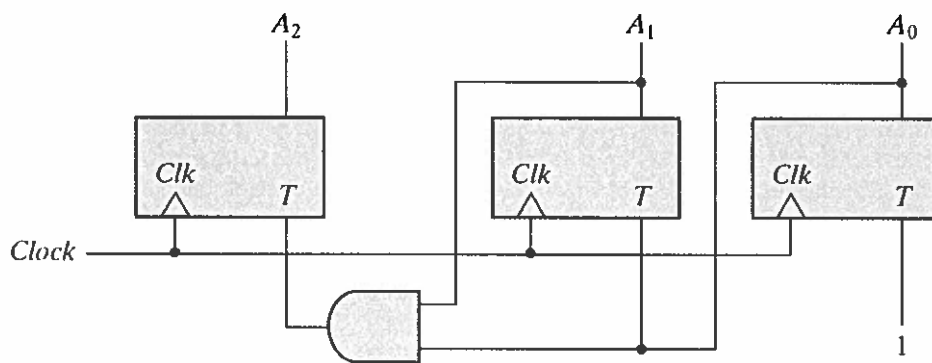


(20 pts) 2. Analysis of Clocked Sequential Logic Circuits

(5 pts) 2.1. Is the following circuit a Mealy or Moore state machines? Please explain the difference between Mealy and Moore state machines.



(15 pts) 2.2. Analysis of Clocked Sequential Circuits: Please based on the following circuit diagram, show (i) State Equations; (ii) State table; and (iii) State diagram, then explain the function.

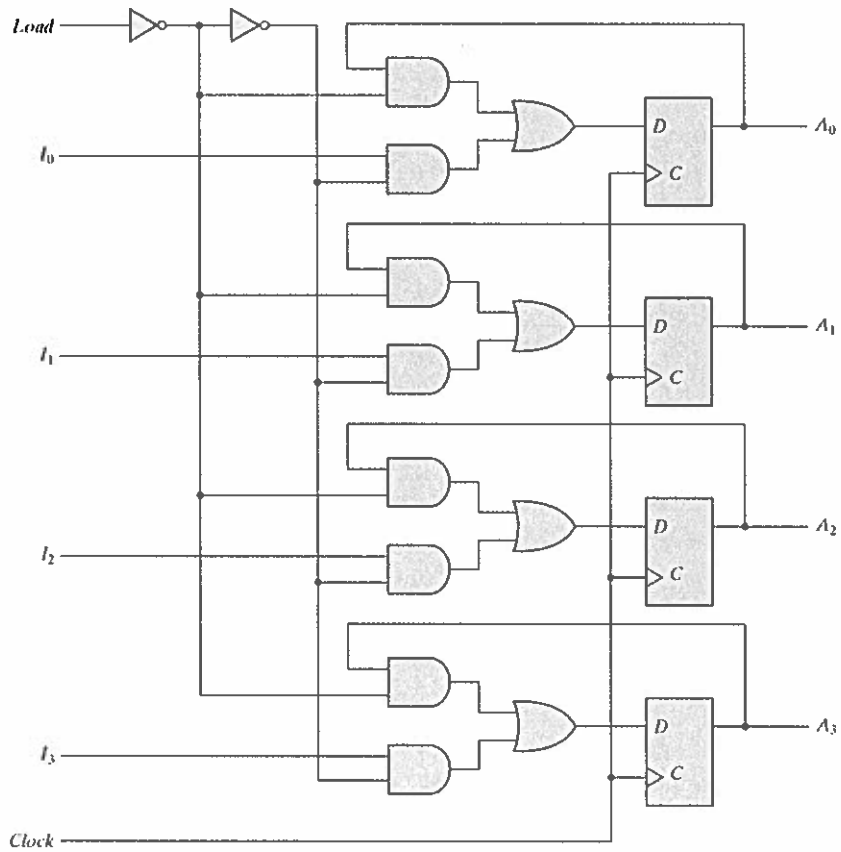


(20 pts)3. Design a sequential logic circuit with two D flip-flops A and B, and an input x.

When $x=1$, the state of the circuit remains the same. When $x=0$, the circuit goes through the state transitions from 00 to 01, to 11, to 10, back to 00, and repeats.

Please demonstrate (i) **state diagram**, (ii) **state table**, and (iii) **then draw the designed sequential circuit**.

(10 pts) 4. Please analyze the following logic circuit, show the state equation, and explain the function of this circuit.



(14pts) 5. Memory: the memory units are specified by the number of words times the number of bits per word.

(4 pts) 5.1 Given a memory consisting of 64k words x 16 bits, (i) **how many address lines** are needed to access the 64k words? (ii) **how many input/output data lines** are needed to read/write information into the memory?

(10 pts) 5.2 Assume a 64k x 16 RAM chip uses coincident decoding by splitting the internal decoder into row selection and column selection. Assume the RAM cell array is square, (i) **what is the size of each decoder?** (ii) **Please draw a diagram to show the two-dimensional decoding structure for the 64k word memory.**

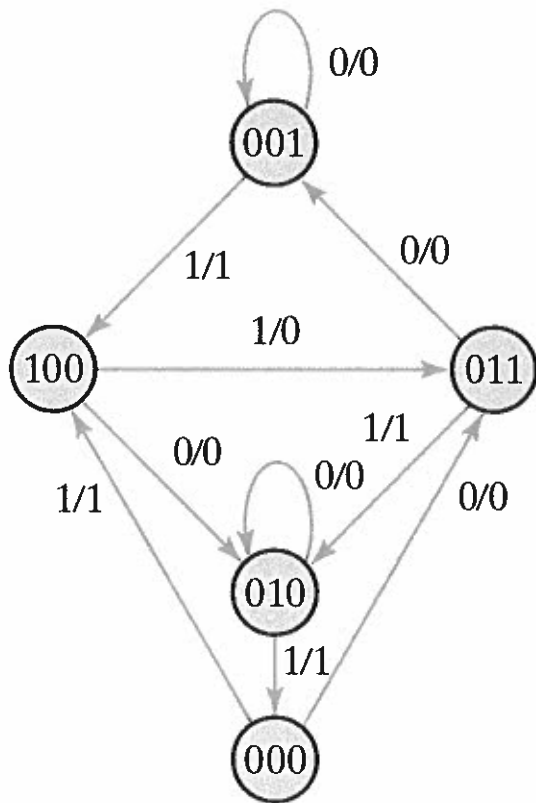
(12 pts) 6. Programmable Logic Array (PLA)

Please using a PLA to implement the following two Boolean functions, considering minimizing the number of product terms. (Please show your implementation process, and finalize with the circuit of the designed PLA)

$$F_1(A,B,C)=\sum(0, 2, 3, 6)$$

$$F_2(A,B,C)=\sum(2, 4, 5, 7)$$

(15 pts) 7. Please design the sequential circuit specified by the following state diagram using **T flip-flops**. Please follow the design procedure and end with the drawing of the sequential circuit.



ABOVE AVERAGE

ELEG 4330 Introduction to Digital Design
Department of Electrical & Computer Engineering
PVAMU

Final Exam

Fall 2021

Name: Tri Heem Neville

1	24	20
2	20	20.5
3	20	19
4	10	9
5	14	9
6	12	12
7	15	15
Total	100+15	103.5

Instructions:

- (1) This exam is close book and notes. One cheatsheet is allowed.
- (2) Cell phone is not allowed.
- (3) Please keep your middle steps in the exam paper. Only the final result will NOT give you full credit.

You have 120 minutes to complete the exam. The points for each problem are indicated. This exam counts for 25% of your course grade.

(24 pts) 1. Sequential digital logic

(4 pts) (1.1). Please briefly compare Combinational logic and Sequential logic.

Sequential logic is based on previous state, input, and output.

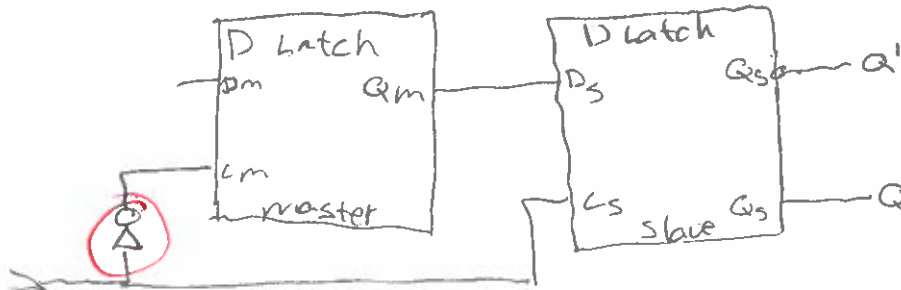
Combinational logic is only based on the input and output.

(4 pts) (1.2). Please briefly compare Synchronous sequential logic circuits and Asynchronous sequential logic circuits.

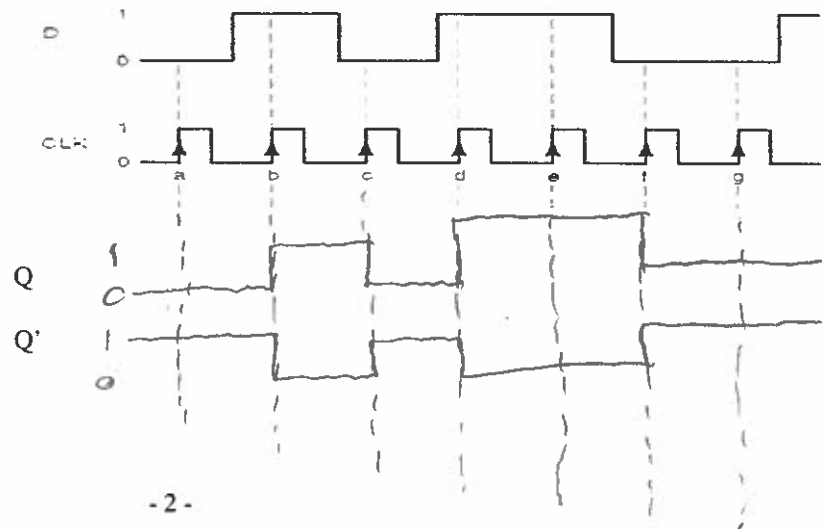
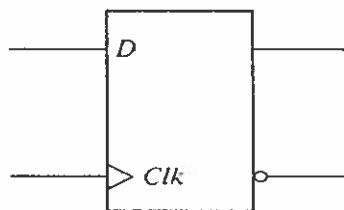
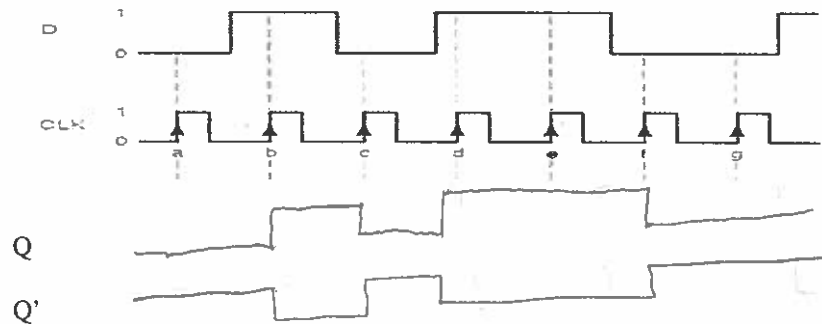
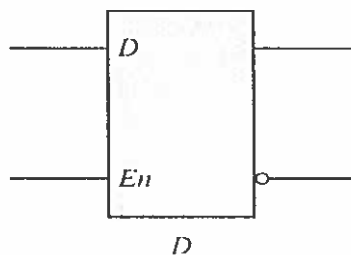
Synchronous circuits: clock input of all FF receive the same common clock

Asynchronous circuit: clock input is different, simple circuit, low power consumption, output change delayed

(6 pts) (1.3). Please demonstrate how to construct a D flip-flop from D Latches (with falling edge triggered).

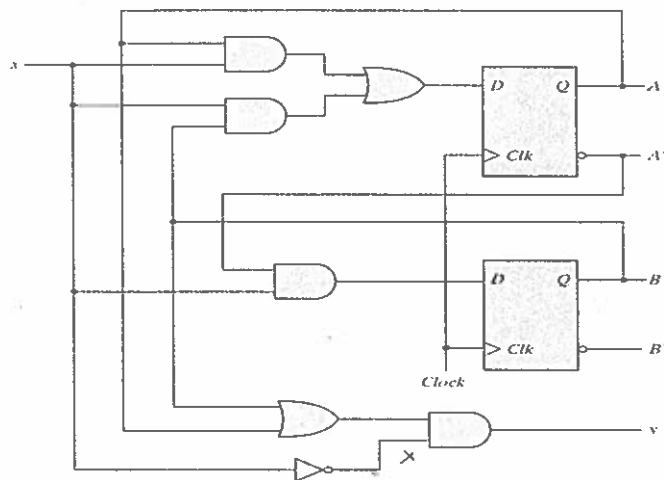


(10pts) (1.4). Given the following timing diagram, please draw the corresponding output(s) in the timing diagram. Assume Q start with 0.



(20 pts) 2. Analysis of Clocked Sequential Logic Circuits

(5 pts) 2.1. Is the following circuit a Mealy or Moore state machines? Please explain the difference between Mealy and Moore state machines.

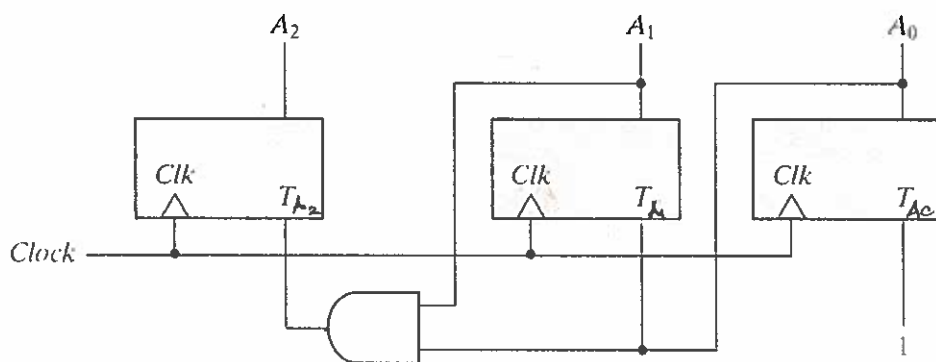


The circuit is a Mealy machine because the output is dependent on the present state and input.

Mealy: output = $f(\text{present state, input})$

Moore: output = $f(\text{present state})$

(15 pts) 2.2. Analysis of Clocked Sequential Circuits: Please based on the following circuit diagram, show (i) State Equations; (ii) State table; and (iii) State diagram, then explain the function.



Input Equations:

$$T_{A2} = A_1 A_0 \quad T_{A1} = A_0$$

$$T_{A0} = 1$$

State Equations:

$$A_2(t+1) = A_2 A_1' + A_2 A_0' + A_2' A_1$$

$$A_1(t+1) = A_1' A_0 + A_1 A_0'$$

$$A_0(t+1) = A_0'$$

State table

P.S			N.S					
A ₂	A ₁	A ₀	A ₂ (t+1)	A ₁ (t+1)	A ₀ (t+1)	T _{A2}	T _{A1}	T _{A0}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	0	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

A ₂	A ₁	A ₀	0	1	1	0
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	1	0	1
0	1	1	0	0	0	1

$$A_2(t+1) = A_2 A_1' + A_2 A_0' + A_2' A_1$$

A ₂	A ₁	A ₀	0	1	1	0
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	1	0	1
0	1	1	0	0	0	1

$$A_1(t+1) = A_1' A_0 + A_1 A_0'$$

A ₂	A ₁	A ₀	0	1	1	0
0	0	0	1	0	0	1
0	0	1	1	0	0	1
0	1	0	1	0	0	1
0	1	1	1	0	0	1

$$A_0(t+1) = A_0'$$

State Diagram!!!

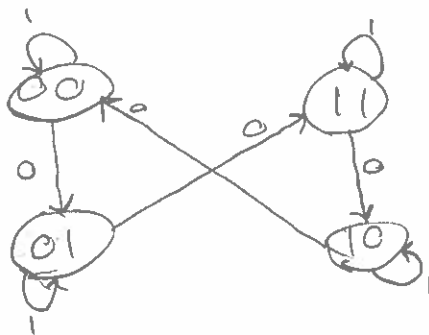
19

(20 pts) 3. Design a sequential logic circuit with two D flip-flops A and B, and an input x.

When $x=1$, the state of the circuit remains the same. When $x=0$, the circuit goes through the state transitions from 00 to 01, to 11, to 10, back to 00, and repeats.

Please demonstrate (i) state diagram, (ii) state table, and (iii) then draw the designed sequential circuit.

i) State Diagram:



ii) State Table:

P.S		Input	Next state	
A	B	X	$A(t+1)$	$B(t+1)$
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

A \ Bx	00	01	11	10
0	0	0	0	1
1	0	1	1	0

$$A(t+1) = Ax + AB + Bx'$$

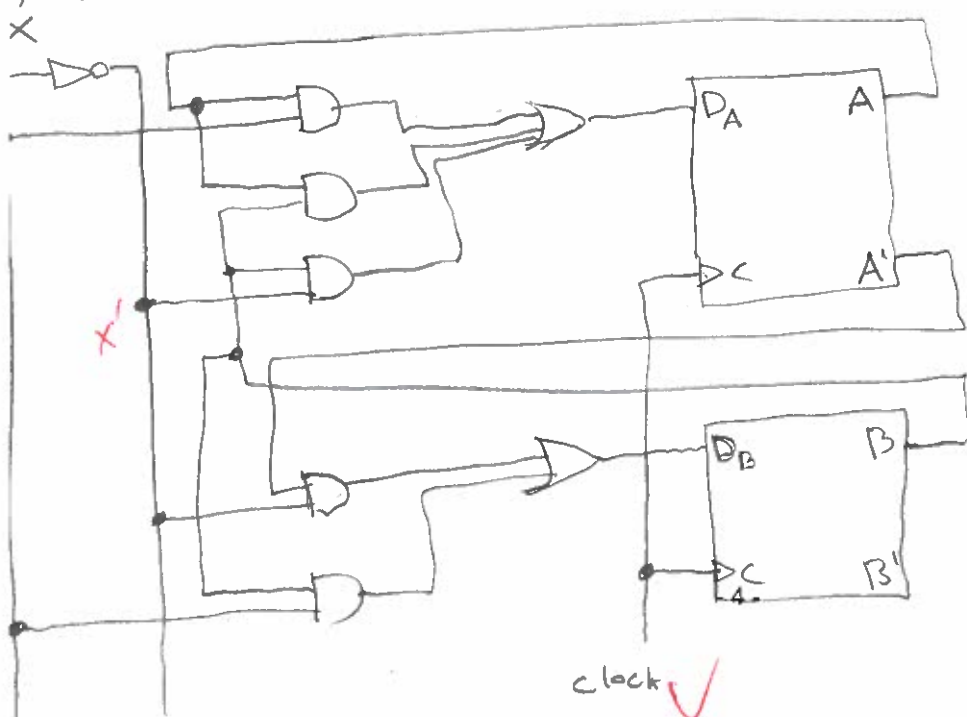
$$D_A = Ax + \underline{AB} + Bx'$$

A \ Bx	00	01	11	10
0	1	0	1	1
1	0	0	1	0

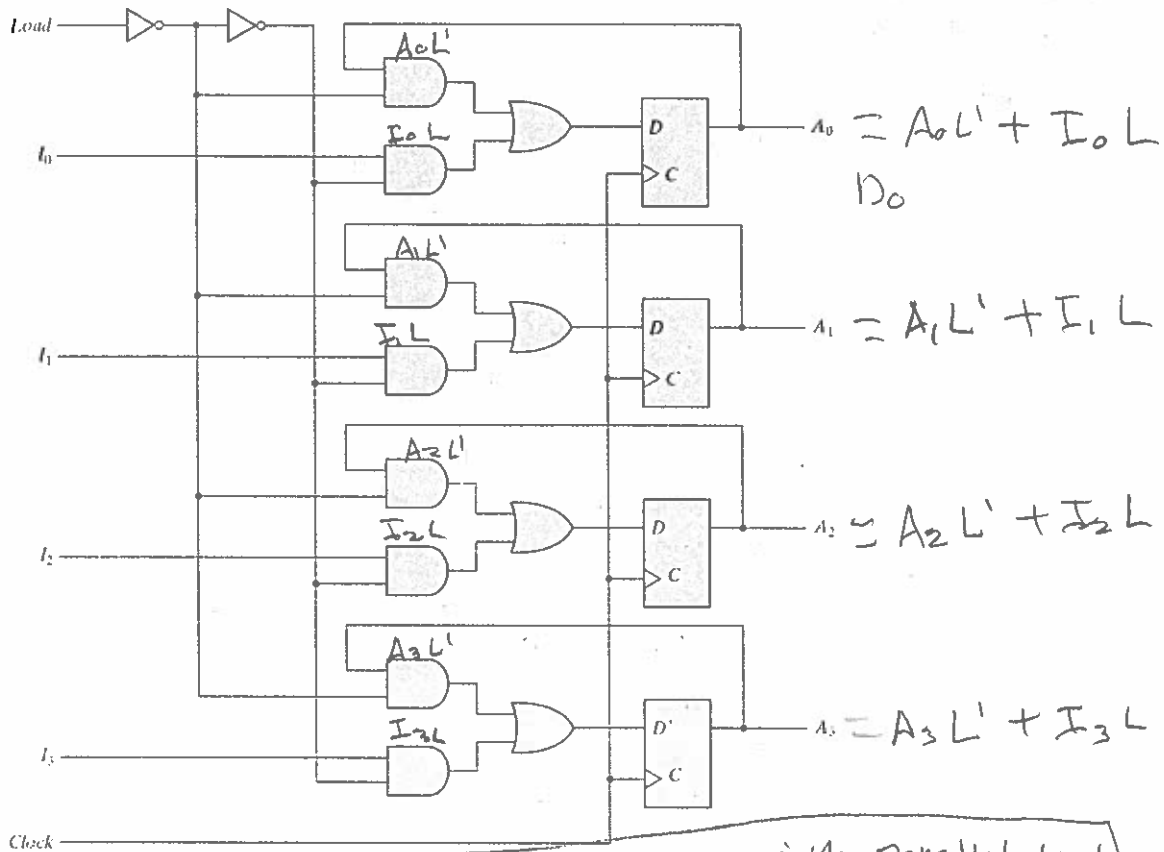
$$B(t+1) = A'x' + Bx$$

$$D_B = A'x' + Bx$$

iii) Sequential Circuit:



(10 pts) 4. Please analyze the following logic circuit, show the state equation, and explain the function of this circuit.



Function of circuit is a 4-bit register with parallel load

State Equation:

$$A_0(t+1) = A_0L' + I_0L$$

$$A_1(t+1) = A_1L' + I_1L$$

$$A_2(t+1) = A_2L' + I_2L$$

$$A_3(t+1) = A_3L' + I_3L$$

(14pts) 5. Memory: the memory units are specified by the number of words times the number of bits per word.

(4 pts) 5.1 Given a memory consisting of 64k words x 16 bits, (i) how many address lines are needed to access the 64k words? (ii) how many input/output data lines are needed to read/write information into the memory?

64k words x 16 bits = $2^6 \times 2^{10} \times 16 = 2^{16} \times 2^4$

A=16 D=16

i) address lines = 16

ii) input/output lines = 16

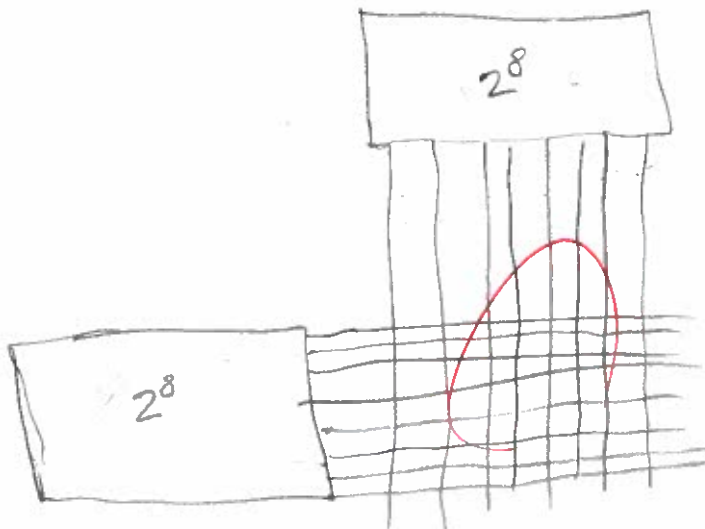
$k = 2^{10}$

(10 pts) 5.2 Assume a 64k x 16 RAM chip uses coincident decoding by splitting the internal decoder into row selection and column selection. Assume the RAM cell array is square, (i) what is the size of each decoder? (ii) Please draw a diagram to show the two-dimensional decoding structure for the 64k word memory.

$64k = 2^6 \times 2^{10} = 2^{16} = 2^8 \times 2^8 = 256 \times 256$

$k = 2^{10}$

i) each decoder has 8 input lines and 256 output lines



(12 pts) 6. Programmable Logic Array (PLA)

12

Please using a PLA to implement the following two Boolean functions, considering minimizing the number of product terms. (Please show your implementation process, and finalize with the circuit of the designed PLA)

$$F_1(A,B,C) = \sum(0, 2, 3, 6)$$

$$F_2(A,B,C) = \sum(2, 4, 5, 7)$$

A \ BC	00	01	11	10
0	1	0	1	1
1	0	1	1	0

$$F_1 = A'B + BC' + A'B'C'$$

$$F_1' = A.B' + AC + B'C$$

A \ BC	00	01	11	10
0				1
1	1	1	1	

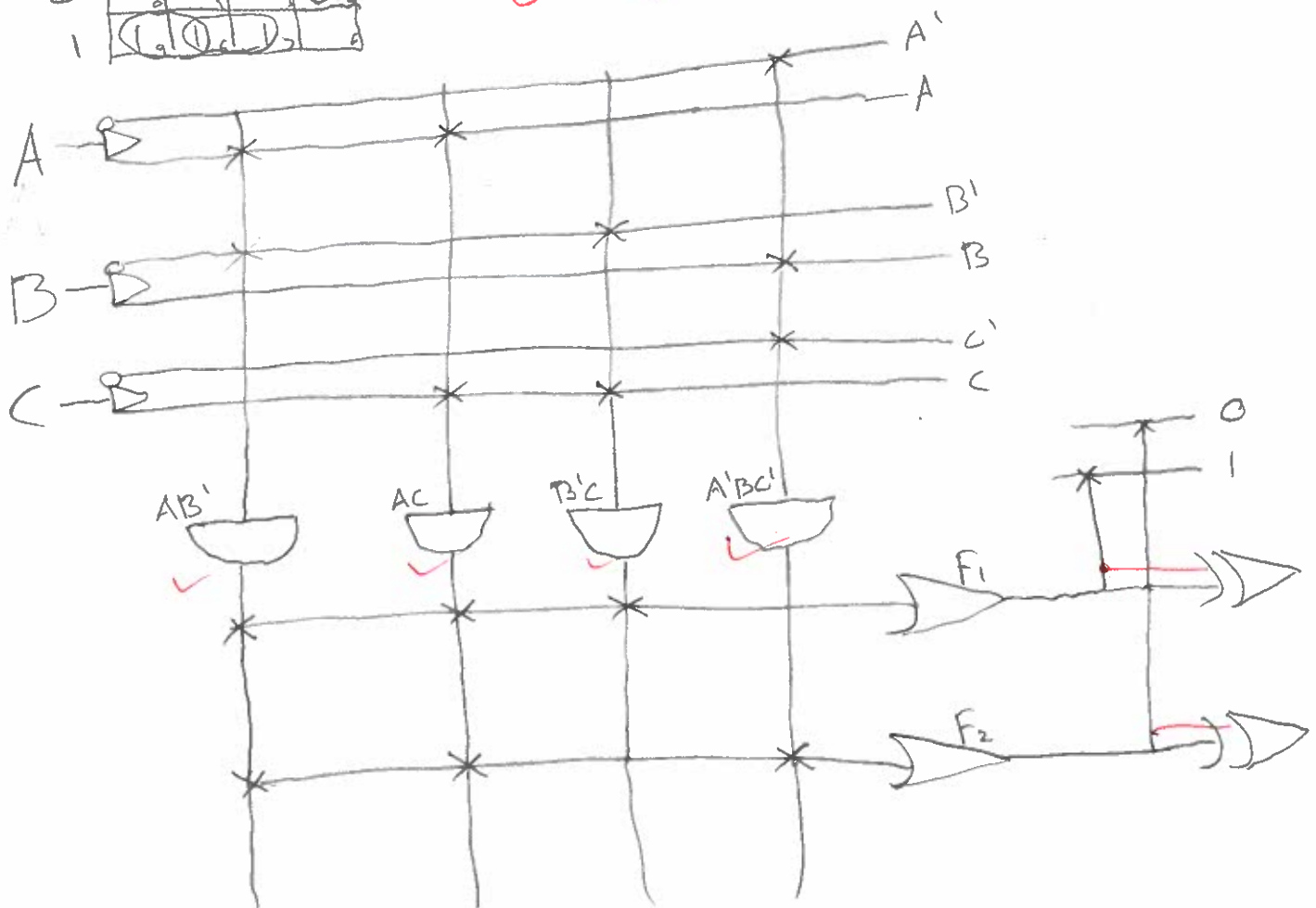
$$F_2 = AB' + AC + A'BC'$$

original:
Product terms: 6

simplified
Product terms: 4

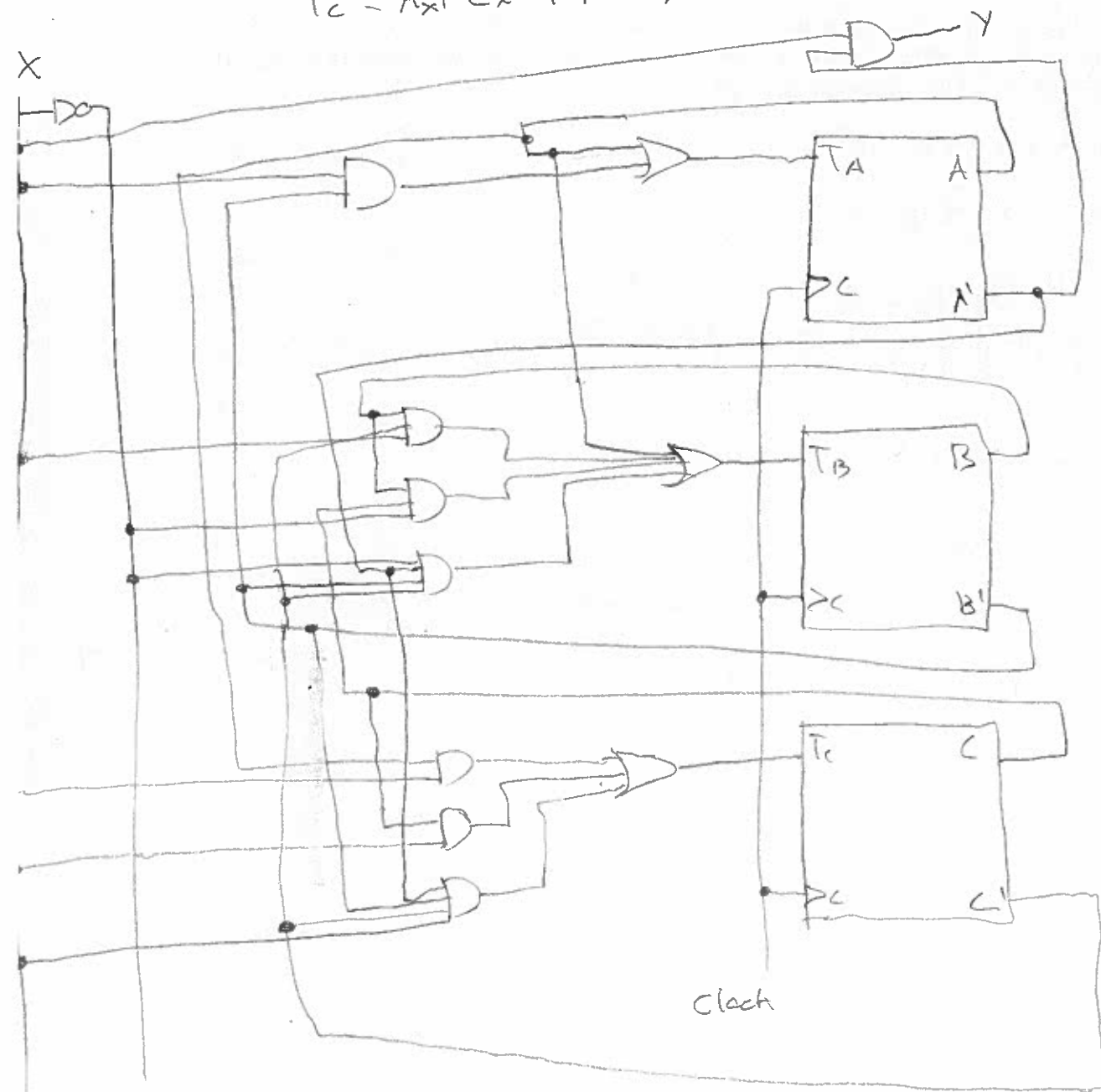
$$1 \oplus x = x'$$

$$0 \oplus x = x$$

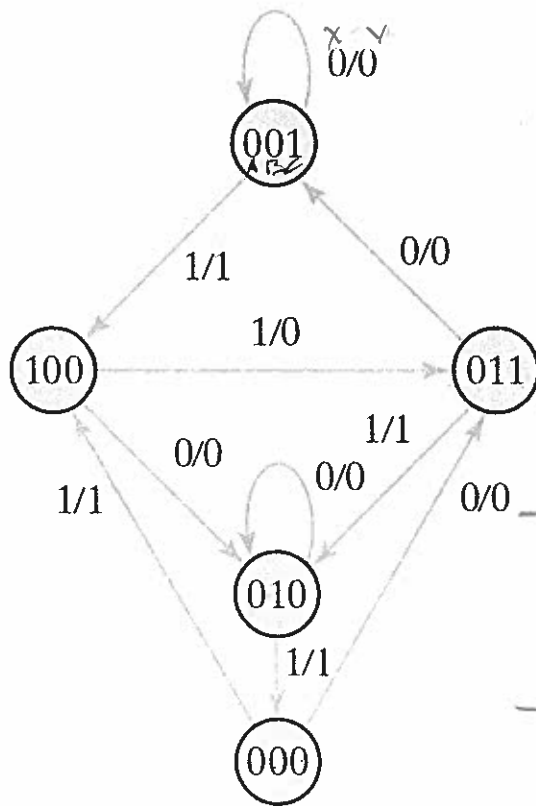


Circuit

$$T_A = A + B'X \quad T_B = A + BC'X + BCX' + A'B'C'X' \\ T_C = Ax + Cx + A'B'C'X' \quad Y = A'X$$



(15 pts) 7. Please design the sequential circuit specified by the following state diagram using T flip-flops. Please follow the design procedure and end with the drawing of the sequential circuit.



State Table

P.S	In	Next State			out	T _A	T _B	T _C
A B C	X	A(t+1)	B(t+1)	C(t+1)	Y			
0 0 0	0	0	1	1	0	0	1	1
0 0 0	1	1	0	0	1	1	0	0
0 0 1	0	0	0	1	0	0	0	0
0 0 1	1	1	0	0	1	1	0	1
0 1 0	0	0	1	0	0	0	0	0
0 1 0	1	0	0	0	1	0	1	0
0 1 1	0	0	0	1	0	0	1	0
0 1 1	1	0	1	0	1	0	0	1
1 0 0	0	0	1	0	0	1	1	0
1 0 0	1	0	1	1	0	1	1	1
1 0 1	0	x	x	x	x	x	x	x
1 0 1	1	x	x	x	x	x	x	x
1 1 0	0	x	x	x	x	x	x	x
1 1 0	1	x	x	x	x	x	x	x
1 1 1	0	x	x	x	x	x	x	x
1 1 1	1	x	x	x	x	x	x	x

AB \ CX	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	x	x	x	x
10	x	x	x	x

$$T_A = A + B'x$$

AB \ CX	00	01	11	10
00	1	0	0	0
01	0	1	0	1
11	x	x	x	x
10	x	x	x	x

$$T_B = A + BC'x + BCx' + \underline{A'B'C'x'}$$

AB \ CX	00	01	11	10
00	1	0	1	0
01	0	0	1	0
11	x	x	x	x
10	0	x	x	x

$$T_C = Ax + Cx + A'B'C'x'$$

AB \ CX	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	x	x	x	x
10	0	0	x	x

$$Y = A'x$$

Circuit on previous back page

AVERAGE

ELEG 4330 Introduction to Digital Design
Department of Electrical & Computer Engineering
PVAMU

Final Exam

Fall 2021

Name: Mohamed Ahmed

1	24	20
2	20	16
3	20	15
4	10	10
5	14	4
6	12	11
7	15	8
Total	100+15	84

Instructions:

- (1) This exam is close book and notes. One cheatsheet is allowed.
- (2) Cell phone is not allowed.
- (3) Please keep your middle steps in the exam paper. Only the final result will NOT give you full credit.

You have 120 minutes to complete the exam. The points for each problem are indicated. This exam counts for 25% of your course grade.

(24 pts) 1. Sequential digital logic

(4 pts) (1.1). Please briefly compare Combinational logic and Sequential logic.

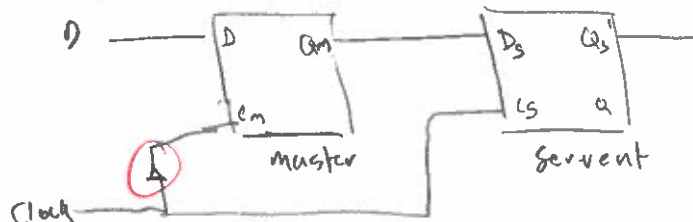
Combinational logic has inputs that can be combined to get output. No Loop.

Sequential logic has both inputs but the output at a FF can go back to get a different input, it also has output.

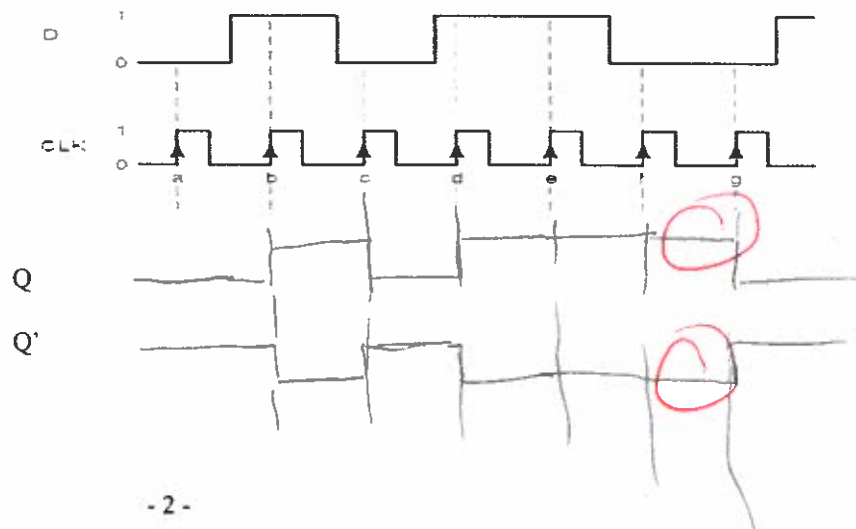
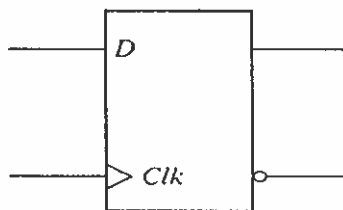
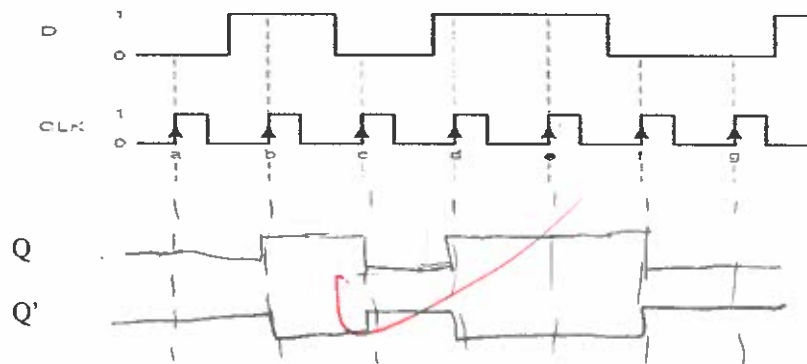
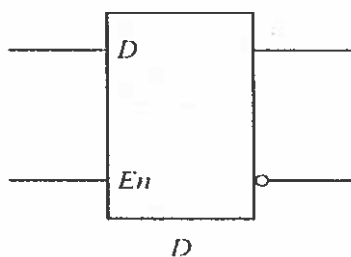
(4 pts) (1.2). Please briefly compare Synchronous sequential logic circuits and Asynchronous sequential logic circuits.

Synchronous logic circuits share the same clock. It is easier to design based on this fact. Asynchronous logic circuits do not share the same clock. The clock pulse comes from other flip flop, each to be on a different clock.

(6 pts) (1.3). Please demonstrate how to construct a D flip-flop from D Latches (with falling edge triggered).

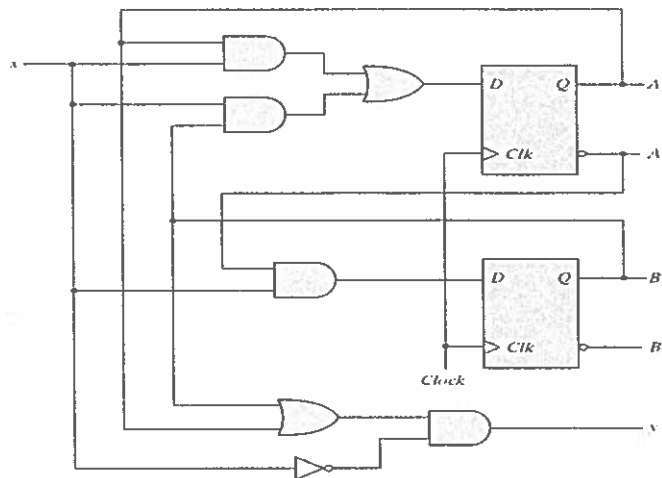


(10pts) (1.4) . Given the following timing diagram, please draw the corresponding output(s) in the timing diagram. Assume Q start with 0.



(20 pts) 2. Analysis of Clocked Sequential Logic Circuits

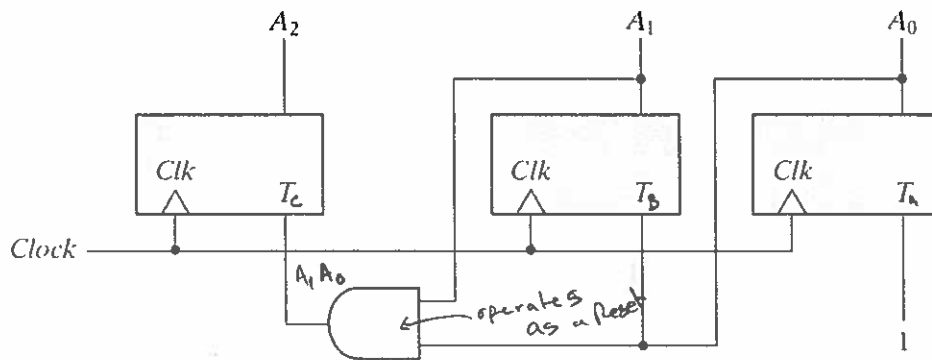
(5 pts) 2.1. Is the following circuit a Mealy or Moore state machines? Please explain the difference between Mealy and Moore state machines.



The main difference between Mealy and Moore state machines is one has input and other has input & output. In this case it is a Moore state Machine



(15 pts) 2.2. Analysis of Clocked Sequential Circuits: Please based on the following circuit diagram, show (i) State Equations; (ii) State table; and (iii) State diagram, then explain the function.



The function is a Counter that reset to zero after 111

Input Eqn $T_A = 1$
 $T_B = A_0$
 $T_C = A_1 A_0$

T	Q(t+1)
0	Q(t)
1	Q'(t)

Characteristic eqn

Present state $A_2 A_1 A_0$	Next state $A_2^+ A_1^+ A_0^+$	T_A	T_B	T_C
0 0 0	0 1 1	1	0	0
0 0 1	0 1 0	1	1	0
0 1 0	0 0 1	1	0	0
0 1 1	1 0 0	1	1	1
1 0 0	1 0 1	1	0	0
1 0 1	1 1 0	1	1	0
1 1 0	1 1 1	1	0	0
1 1 1	0 0 0	1	1	1



$A_1 A_0$	00	01	11	10
A_2	0	0	1	0
1	1	1	0	1

$A_1 A_0$	00	01	11	10
A_2	0	0	0	0
1	1	1	1	1

$A_1 A_0$	00	01	11	10
A_2	1	0	0	1
1	1	0	0	1

$$A_2(t+1) = A_2 A_1' A_0' + A_2 A_1' A_0 + A_2 A_1 A_0' + A_2 A_1 A_0$$

$$A_2(t+1) = A_2 A_1 A_0' + A_2 A_1' A_0 + A_2 A_1 A_0$$

$$A_0(t+1) = A_0'$$

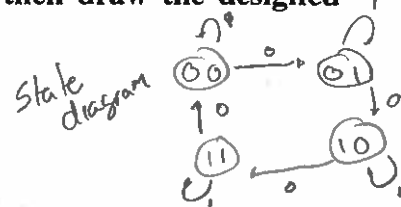
(20 pts) 3. Design a sequential logic circuit with two D flip-flops A and B, and an input x.

When $x=1$, the state of the circuit remains the same. When $x=0$, the circuit goes through the state transitions from 00 to 01, to 11, to 10, back to 00, and repeats.

Please demonstrate (i) state diagram, (ii) state table, and (iii) then draw the designed sequential circuit.

State Table

Present State		Input	Next State		Flip flops	
A	B		A(t+1)	B(t+1)	D_A	D_B
0	0	0	0	1	0	1
0	0	1	0	0	0	0
0	1	0	1	0	1	0
0	1	1	0	1	0	1
1	0	0	1	1	1	1
1	0	1	1	0	1	0
1	1	0	0	0	0	0
1	1	1	1	1	1	1



$$D_A = A(t+1)$$

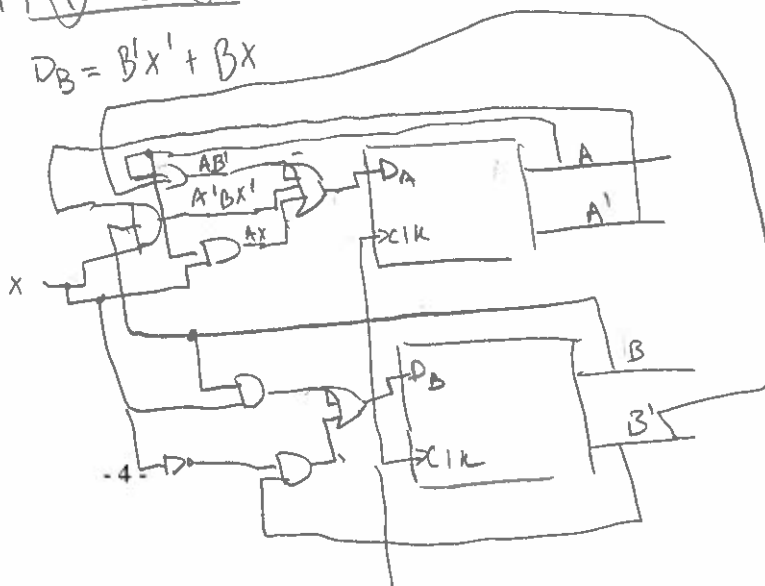
A \ Bx	00	01	11	10
0	0	0	0	1
1	1	1	1	0

$$D_A = A'Bx' + AB' + Ax$$

$$D_B = B(t+1)$$

A \ Bx	00	01	11	10
0	1	0	1	0
1	1	0	1	0

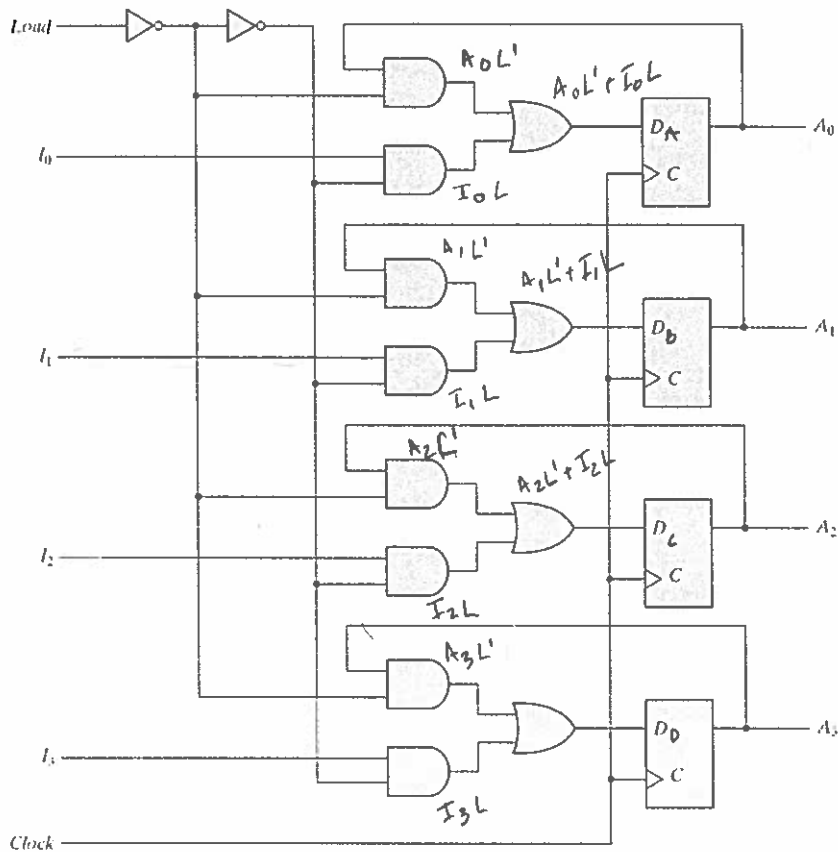
$$D_B = B'x' + Bx$$



10

(10 pts) 4. Please analyze the following logic circuit, show the state equation, and explain the function of this circuit.

$$\text{Load} = L$$



$$D_A = A_0 L' + I_0 L$$

$$D_B = A_1 L' + I_1 L$$

$$D_C = A_2 L' + I_2 L$$

$$D_D = A_3 L' + I_3 L$$

When $L=1$

$$D_A = I_0$$

$$D_B = I_1$$

$$D_C = I_2$$

$$D_D = I_3$$

The circuit behaves as a register, $Q(t+1) = D$

$$Q(t+1) = D$$

When $L=0$

$$D_A = A_0$$

$$D_B = A_1$$

$$D_C = A_2$$

$$D_D = A_3$$

(14pts) 5. Memory: the memory units are specified by the number of words times the number of bits per word.

(4 pts) 5.1 Given a memory consisting of 64k words x 16 bits, (i) how many address lines are needed to access the 64k words? (ii) how many input/output data lines are needed to read/write information into the memory?

$$1K = 1024 \text{ bits}$$

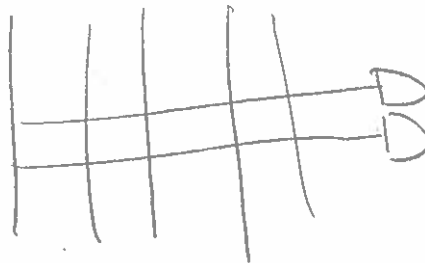
$$64K \times \frac{1024 \text{ bits}}{1K} = 65536 = 2^{16} \times 16$$

$$A = 16 \quad D = 13$$

(10 pts) 5.2 Assume a 64k x 16 RAM chip uses coincident decoding by splitting the internal decoder into row selection and column selection. Assume the RAM cell array is square, (i) what is the size of each decoder? (ii) Please draw a diagram to show the two-dimensional decoding structure for the 64k word memory.

i) Decoder will be 13

ii)



(12 pts) 6. Programmable Logic Array (PLA)

Please using a PLA to implement the following two Boolean functions, considering minimizing the number of product terms. (Please show your implementation process, and finalize with the circuit of the designed PLA)

$$F_1(A,B,C) = \sum(0, 2, 3, 6)$$

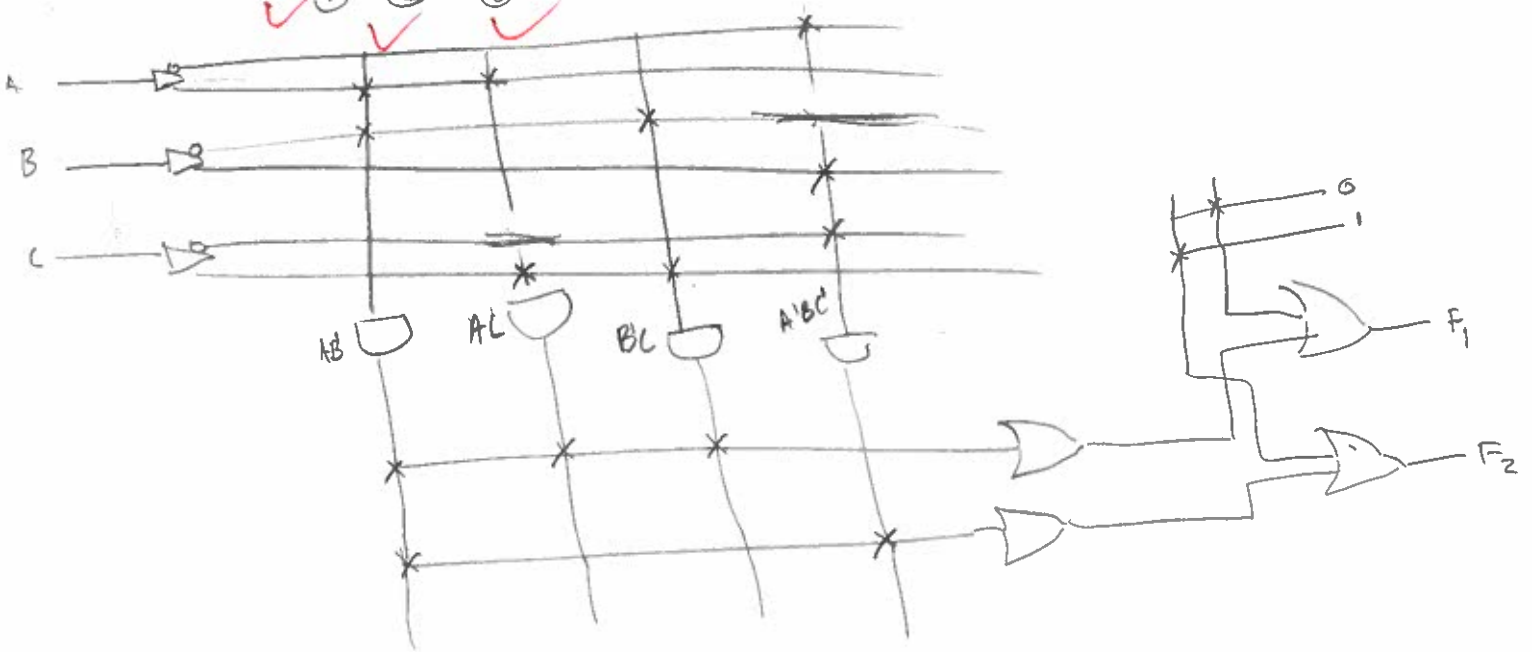
$$F_2(A,B,C) = \sum(2, 4, 5, 7)$$

A \ BC	00	01	11	10
0	1	1	1	1
1	1	1	1	1

A \ BC	00	01	11	10
0	0	0	0	1
1	1	1	0	0

$$F_2 = \underline{A}B' + A'B\underline{C}' \quad (4)$$

$$F_1 = \underline{A}B' + \underline{A}C + B'\underline{C} \quad (1) \quad (2) \quad (3)$$



8

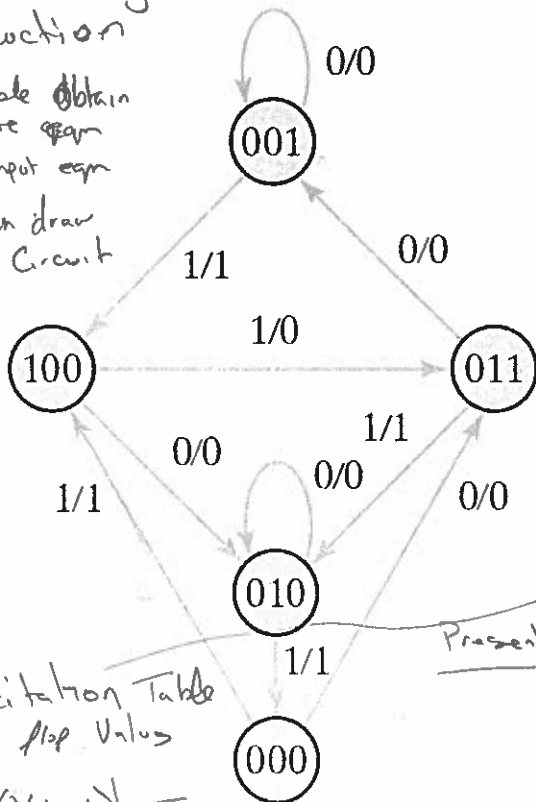
(15 pts) 7. Please design the sequential circuit specified by the following state diagram using **T flip-flops**. Please follow the design procedure and end with the drawing of the sequential circuit.

Steps -> ① State diagram to state table

② State Reduction

③ From State Table obtain
1) State eqn
2) Input eqn

④ From input eqn draw Sequential Circuit



Using Excitation Table to get Flip Flop Values

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

Present State	Input	Next State	Output
ABC	x	$A^+B^+C^+$	y
000	0	011	0
001	1	100	1
010	0	001	0
011	1	100	1
100	0	010	0
101	1	000	1
110	0	001	0
111	1	010	1
x 100	0	010	0
100	1	011	0

State Reduction

100 = 010

Present State	Input	Next State	Output	T_A	T_B	T_C
ABC	x	$A^+B^+C^+$	y			
000	0	011	0	0	1	1
001	0	001	0	0	0	0
010	1	000	1	0	0	0
011	0	001	0	0	0	0
100	1	010	1	0	0	1
101	1	011	0	0	1	1

T_A	AB	00	01	11	10
00					
01					
11					
10					

T_B	AB	00	01	11	10
00					
01					
11					
10					

T_C	AB	00	01	11	10
00					
01					
11					
10					

BELOW AVERAGE

ELEG 4330 Introduction to Digital Design
Department of Electrical & Computer Engineering
PVAMU

Final Exam

Fall 2021

Name: Gabriel Maxwell Jr

1	24	10
2	20	19
3	20	10
4	10	8
5	14	11
6	12	3
7	15	7
Total	100+15	68

Instructions:

- (1) This exam is close book and notes. One cheatsheet is allowed.
- (2) Cell phone is not allowed.
- (3) Please keep your middle steps in the exam paper. Only the final result will NOT give you full credit.

You have 120 minutes to complete the exam. The points for each problem are indicated. This exam counts for 25% of your course grade.

(24 pts) 1. Sequential digital logic

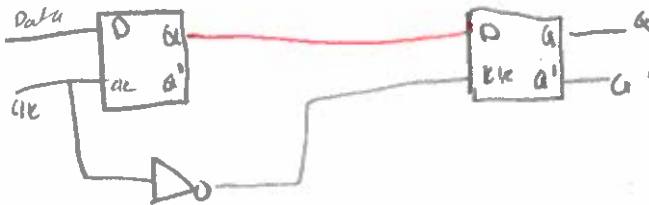
(4 pts) (1.1). Please briefly compare Combinational logic and Sequential logic.

4
Combinational & Sequential circuits are both logic circuits but Combinational circuits depend on present inputs only and have no memory element. Sequential circuits outputs depend on both present & past inputs & has a memory element

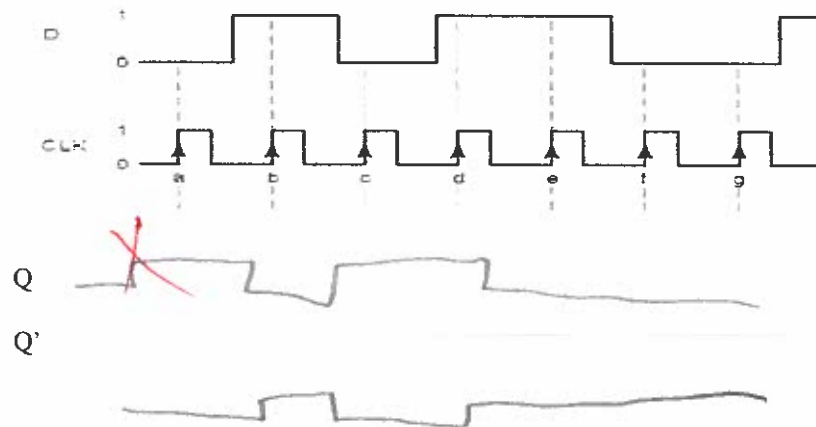
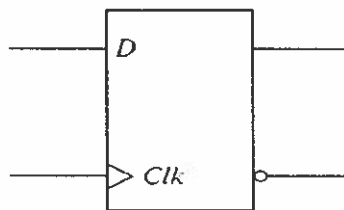
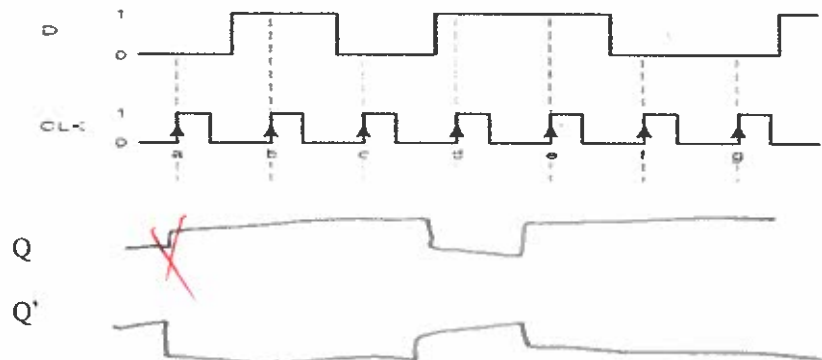
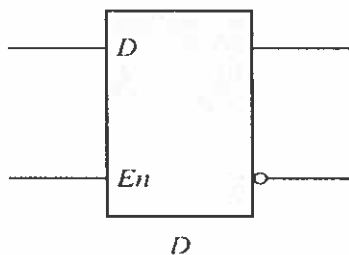
1
(4 pts) (1.2). Please briefly compare Synchronous sequential logic circuits and Asynchronous sequential logic circuits.

Synchronous circuits there is no propagation delay and in asynchronous circuits there is a propagation delay from flip flop to flip flop

3
(6 pts) (1.3). Please demonstrate how to construct a D flip-flop from D Latches (with falling edge triggered).



1
(10pts) (1.4). Given the following timing diagram, please draw the corresponding output(s) in the timing diagram. Assume Q start with 0.

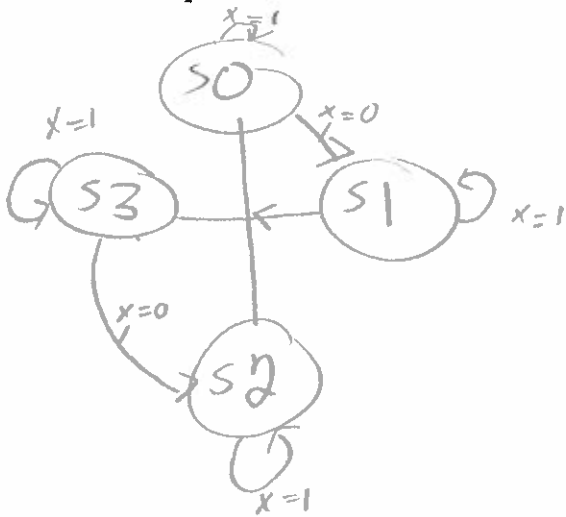


10

(20 pts) 3. Design a sequential logic circuit with two D flip-flops A and B, and an input x.

When $x=1$, the state of the circuit remains the same. When $x=0$, the circuit goes through the state transitions from 00 to 01, to 11, to 10, back to 00, and repeats.

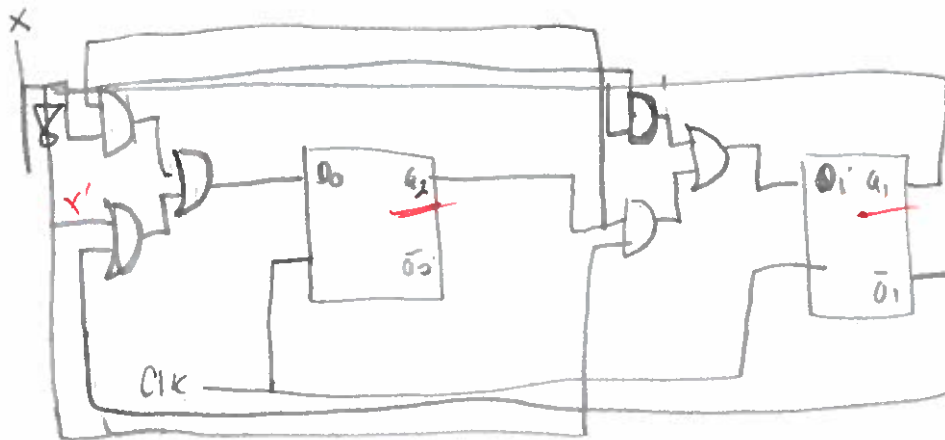
Please demonstrate (i) state diagram, (ii) state table, and (iii) then draw the designed sequential circuit.



state table

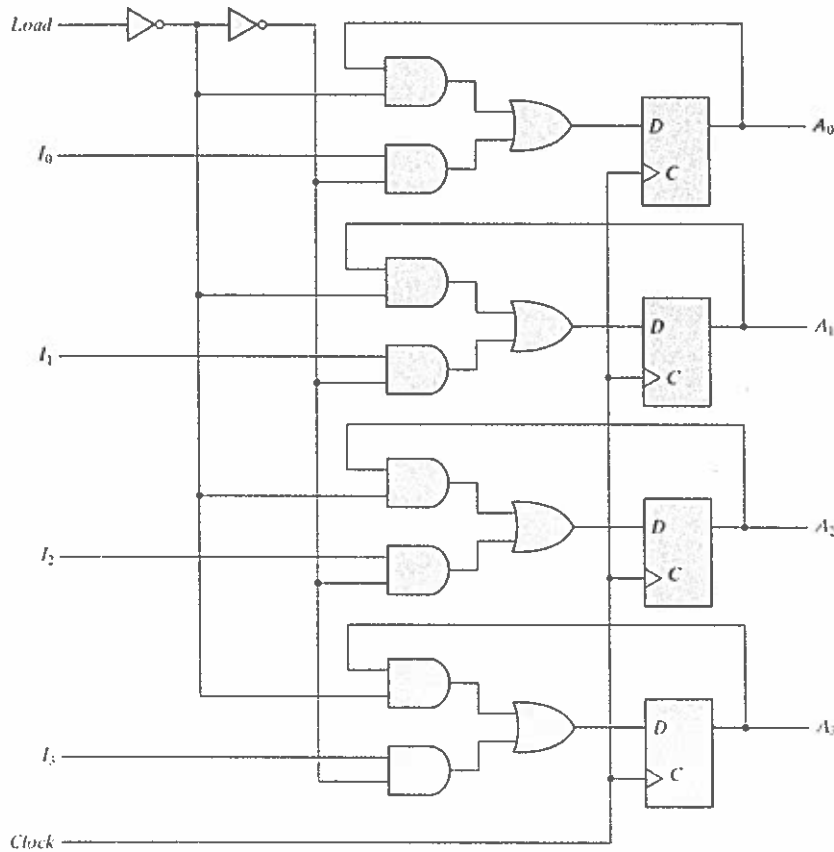
PS	IN	NS
S0	1	S0
S0	0	S1
S1	1	S1
S1	0	S3
S2	1	S2
S2	0	S0
S3	1	S3
S3	0	S2

normal one.



equation?

(10 pts) 4. Please analyze the following logic circuit, show the state equation, and explain the function of this circuit.



state equation

$$DA(t) = \text{load}(A_0) + I_0 \text{ load}$$

$$DB = \text{load}(A_1) + I_1 \text{ load}$$

$$DC = \text{load}(A_2) + I_2 \text{ load}$$

$$DD = \text{load}(A_3) + I_3 \text{ load}$$

I_0, I_1, I_2, I_3 are the i/p

$$\text{Load} = 0, DA = A_0$$

$$DB = A_1$$

$$DC = A_2$$

$$DD = A_3$$

$$\text{Load} = 1, I_0, I_1, I_2, I_3 \text{ are } A_0, A_1, A_2, A_3$$

Load = 1 1/p of flops will be into DFF

Load = 0 1/p of flops will be into D, FF

(14pts) 5. Memory: the memory units are specified by the number of words times the number of bits per word.

(4 pts) 5.1 Given a memory consisting of 64k words x 16 bits, (i) how many address lines are needed to access the 64k words? (ii) how many input/output data lines are needed to read/write information into the memory?

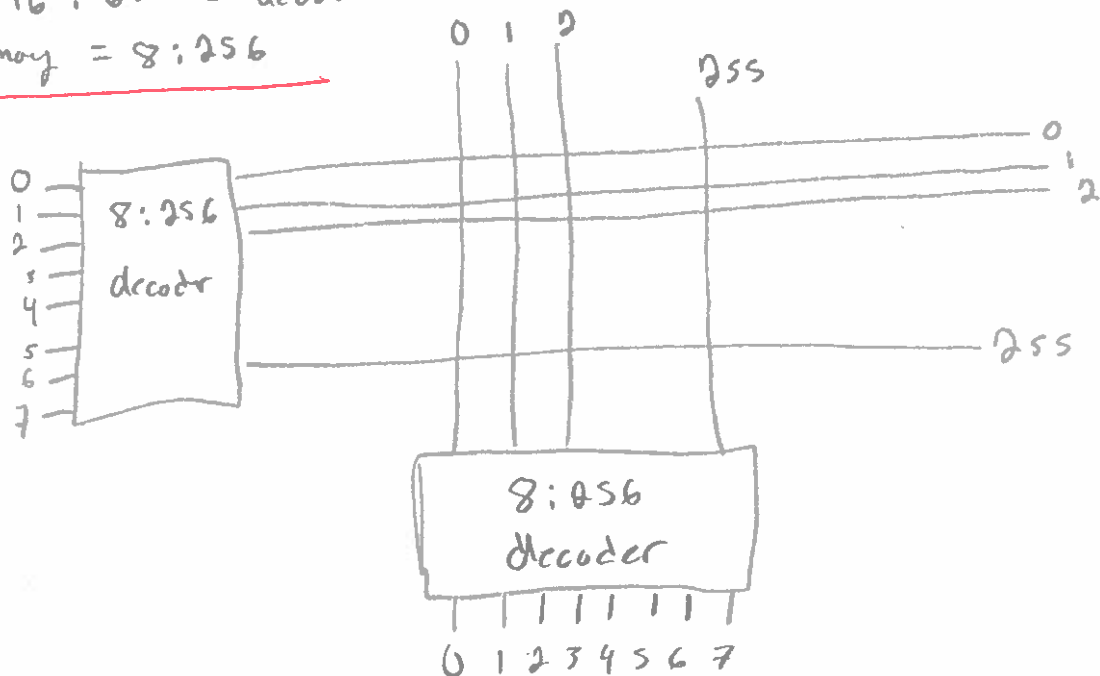
$$\log_2(64) = 6 \text{ number of address lines}$$

16 is the size of each word + data line

(10 pts) 5.2 Assume a 64k x 16 RAM chip uses coincident decoding by splitting the internal decoder into row selection and column selection. Assume the RAM cell array is square, (i) what is the size of each decoder? (ii) Please draw a diagram to show the two-dimensional decoding structure for the 64k word memory.

$i = 16 : 65536 \text{ decoder}$

Memory = 8:256



(12 pts) 6. Programmable Logic Array (PLA)

Please using a PLA to implement the following two Boolean functions, considering minimizing the number of product terms. (Please show your implementation process, and finalize with the circuit of the designed PLA)

$$F_1(A,B,C) = \sum(0, 2, 3, 6)$$

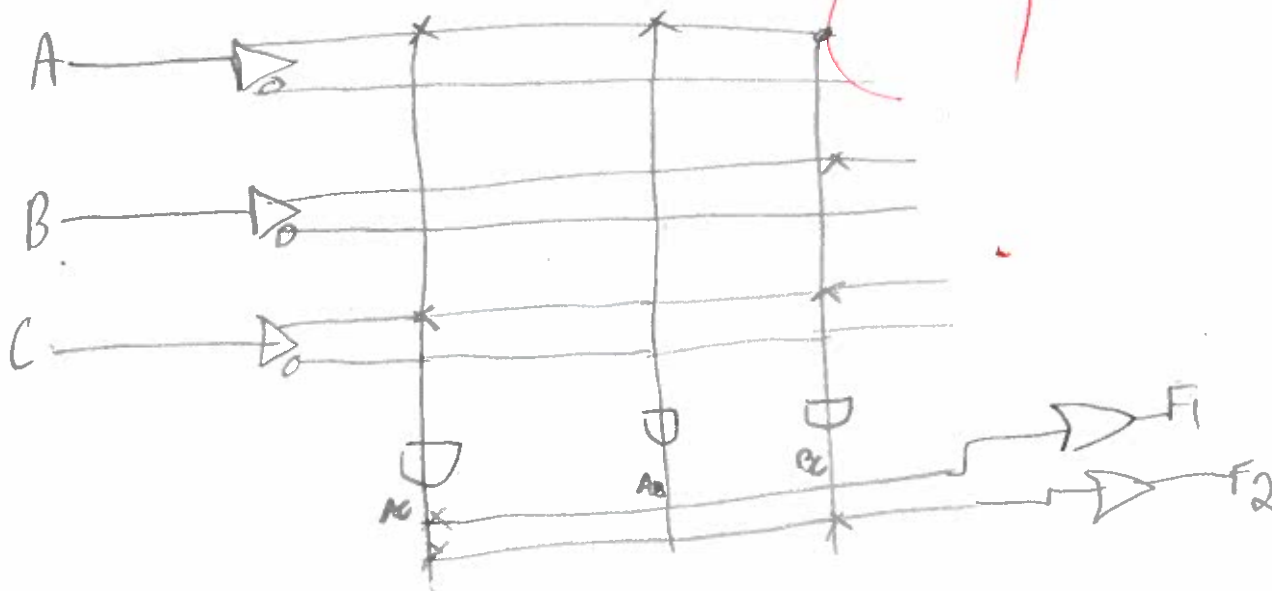
$$F_2(A,B,C) = \sum(2, 4, 5, 7)$$

0	1	3	2
4	5	6	7

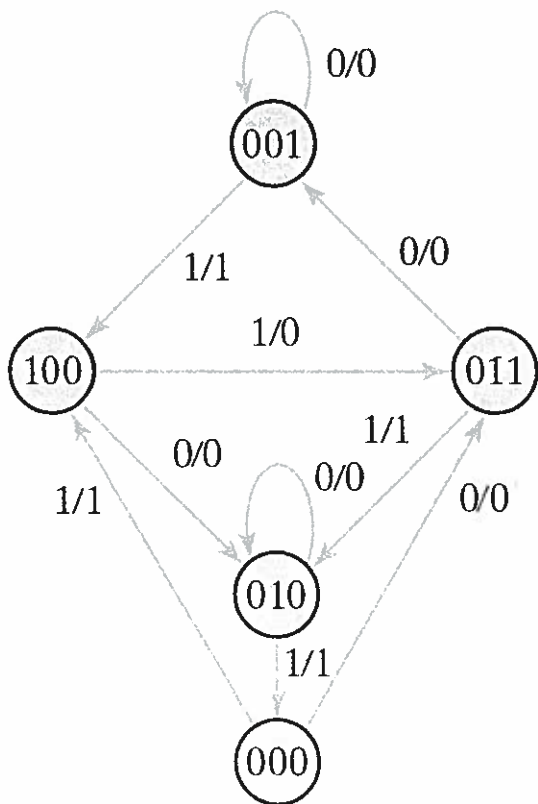
$$F_1 = AC +$$

$$F_2 = AC + BC$$

3 = variables



(15 pts) 7. Please design the sequential circuit specified by the following state diagram using T flip-flops. Please follow the design procedure and end with the drawing of the sequential circuit.



$q_3 q_2 q_1$	x	$q_3^+ q_2^+ q_1^+$	$T_1 T_2 T_3$	y
000	0	011	011	0
000	1	100	000	1
001	0	001	000	0
001	1	100	101	1
010	0	010	000	0
010	1	000	101	1
011	0	001	000	0
011	1	010	010	1
100	0	010	010	0
100	1	010	000	1
101	0	011	110	0
101	1	011	111	0
110	0	xxx	xxx	x
110	1	xxx	xxx	x
111	0	xxx	xxx	x
111	1	xxx	xxx	x

T_3

0	1	1	0
0	0	0	0
x	x	x	x
1	1	x	x

$T_3 = q_3 + \bar{q}_1$

T_3

0	1	1	0
0	0	0	0
x	x	x	x
1	1	x	x

$T_3 = q_3 + q_1$

T_1

1	0	1	0
0	0	1	0
x	x	x	x
0	1	x	x

$T_1 = (q_1 + q_3)x + \bar{q}_3 \bar{q}_2 \bar{q}_1 \bar{x}$

T_1

1	0	1	0
0	0	1	0
x	x	x	x
0	1	x	x

$T_1 = (q_3 + q_1)x + \bar{q}_3 \bar{q}_2 \bar{q}_1 \bar{x}$

